International Journal of Future Generation Communication and Networking
Vol. 13, No. 3s, (2020), pp. 632–637

Pandemic Security System for Police using Neural Networks

K.B Wane [1], Dr. Rahul G. Mapari[2] and Ajin Abraham[3]

[1]*Dept of Electronic and Telecommunication, Pimpri Chinchwad College of Engineering and Research. Pune- 412101, India*

[2] *Dept of Electronic and Telecommunication, Pimpri Chinchwad College of Engineering and Research. Pune- 412101, India*

[3] *Dept of Electronic and Telecommunication, Pimpri Chinchwad College of Engineering and Research. Pune- 412101, India*

[1]*kishore.wane@pccoer.in*, [2]*rahul.mapari@pccoer.in*, [3]*ajinabraham605@gmail.com*

Abstract

During this global pandemic named COVID-19 where social distancing is playing a vital role in preventing the spread of this virus among the people. Even after a strong ordinance given by the government to perform a complete lockdown, citizens are being reckless and showing up on the streets. The police are patrolling the streets round the clock to avoid this situation by risking their own life. So to bring ease to their work and also to keep them safe, we have implemented a new system where we will be using a CCTV camera as a medium to detect whether a set of people are gathering in a certain place and inform the map coordinates of that place to the police control station. This will prevent a social gathering of more than 5 people in a place and help us to fight this pandemic by safeguarding the life of people and also the police officers risking their life. For detection, we will be using one the famous technique of Convolution Neural Networking named YOLO. Through this method, we will be detecting the object(person). Once the detection is done, through certain mathematical calculations we will detect the distance between an object by keeping one object as a reference object. Once the distance between them is less than the threshold set there will be an emergency message sent to the police officials which will contain the coordinates of the location and they can prevent this kind of gathering without actually patrolling on the streets.

*Keywords**:** COVID-19, Pandemic, Neural Network, CCTV, Map Coordinates, Social Distancing*

I. INTRODUCTION

This CNN deep learning model for processing data that includes a grid pattern, like images, which is inspired by the animal visual and designed to automatically and adaptively learn spatial hierarchies of features, from low to high-level patterns [4]. YOLO, a brand-new approach to object detection. Earlier all the work on object detection repurposed classifiers to perform detection. Instead, we frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities, one neural network predicts bounding boxes and sophistication probabilities directly from full images in one evaluation. Since the entire detection pipeline could be a single network, it may be optimized end-to-end directly on detection performance. Our unified architecture is extremely fast [7]. YOLO model processes images in real-time at a mere 45 frames per second. A smaller version of the network, Fast YOLO, processes an astounding 155 frames per second while still achieving double the Mean Average Precision (MAp) of other real-time detectors. Compared to state-of-the-art detection systems, YOLO makes more localization

ISSN: 2233-7857 IJFGCN

errors but is much less likely to predict false detections where nothing exists. Finally, YOLO learns very general representations of objects. It outperforms all other detection methods, including DPM and R-CNN, by a good margin when generalizing from natural images to the artwork on dataset given [9]. In the initial stage we augment the data. In our dataset we have obtained a set of images where there is a total of 3000 training images, 1000 validation image set and 1500 test image set. In data augmentation we also rescale the set of images with a common factor and resize it to 128*128*3. We keep the sheer range to 0.2 and zoom range to 0.2. We also randomly adjust the exposure and saturation of the image by up to an element of 1.5 within the HSV color space.[5]

The layers are also made as follows. The first layer is a layers where we detect the features by keeping 32 filters of 3*3 size having activation function as Relu and also providing an zero padding to the image set. We also use a convolution stride of 2. In the next layer we use a max pooling layer having a filter size of 2*2 and this filter will take the maximum pixel value that contains maximum feature. Again, we take another layer having the same feature but with number of 64 filter as feature detector. We now use a series of more 2 layers having filter 128,256 and 364. Now we add dense having first 256, 256,128 and 64 nodes in respective hidden layers with activation Relu to increase the non-linearity. At the end we have a dense of 1 node with sigmoid as an activation function.

Our learning rate schedule is as follows: within the first epoch, we slowly increase the training rate from 103 to 102. For instance, if we choose to start at a high learning rate our model often diverges because of unstable gradients. We continue training with 102 for 75 epochs, then decrease to 103 for 30 epochs, and finally decrease again to 104 for 30 epochs. To avoid overfitting we use dropout. A dropout layer with rate = 0.5 after the first connected layer prevents co-adaptation between layers.

II. PROPOSED WORK

As in this global crisis of COVID-19 has made sure that social distancing is the only preventive measure that we can apply to avoid a massive spread of this virus among people. This is also suggested by World Health Organization (WHO). The police are trying to maintain this situation by not letting a huge mob gathering in a particular area. So to help the police we have implemented a new system that will help the police officials to patrol the streets without actually being there. So anew detection algorithm has been introduced by us as mentioned in Fig 1.

The detection here is done by using the centroid tracking mechanism where a different id is assigned to an object having different centroid value. This will make sure that the there is no redundancy of same object been recorded multiple times. Here we take the distance from previous centroid calculated by the bounding boxes till the centroid distance from upcoming frame. Once the Id is assigned then the count gets incremented. Now the count will only get incremented if and only if the distance between the object if less that 1 meter. If the count increases more than 5 the coordinates of that location will be given to the police officials and then they can personally survey the location and avoid the gathering.

ISSN: 2233-7857 IJFGCN

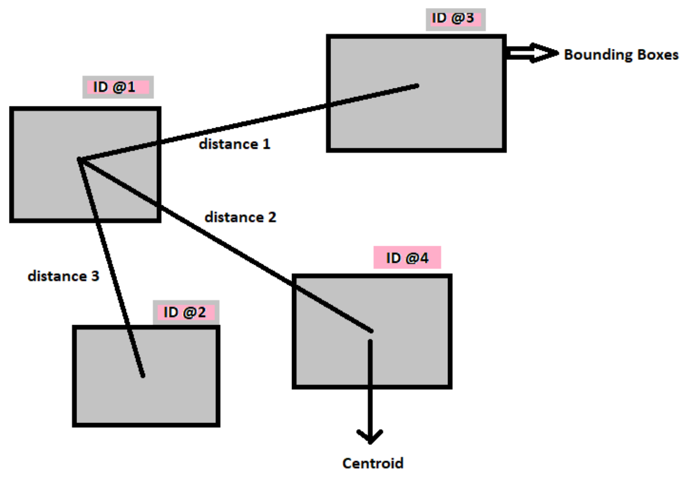

Figure 2. Distance Detection Algorithm using Centroid as Reference

2.1. Flow Chart

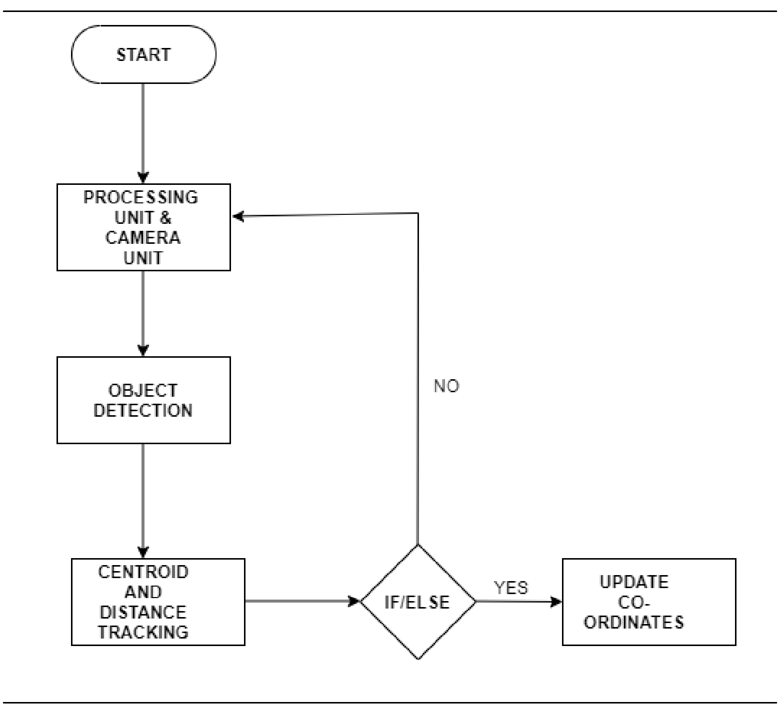

Figure 2. Flow Chart

We start the system by initiating the Processing unit and camera initialization. Then we run our algorithm. Once the object is detected we perform centroid calculation of that object by applying centroid formula on the bounding boxes which is created through image localization. Once the centroid is calculated, we assign a new Id to each centroid and calculate the distance between the centroids. If the centroid distance is above the threshold value then the coordinates of the location is updated and sent to the control stations else it will again move to the processing units and the cycle will move continuously.

ISSN: 2233-7857 IJFGCN

III. RESULTS

A total of 3000 training images and 1000 validation images were used to develop custom weights. The images had to be preprocessed just like the train set by which noise and uniform illumination was obtained. The prepared model showed the mAP of 57.8.

Table 1. Algorithm Comparison

Sr No.	Algorithm	MAp	Time
1	YOLO	57.8	51ms
2	SSD321	45.4	61ms
3	R-FCNN	51.9	81ms

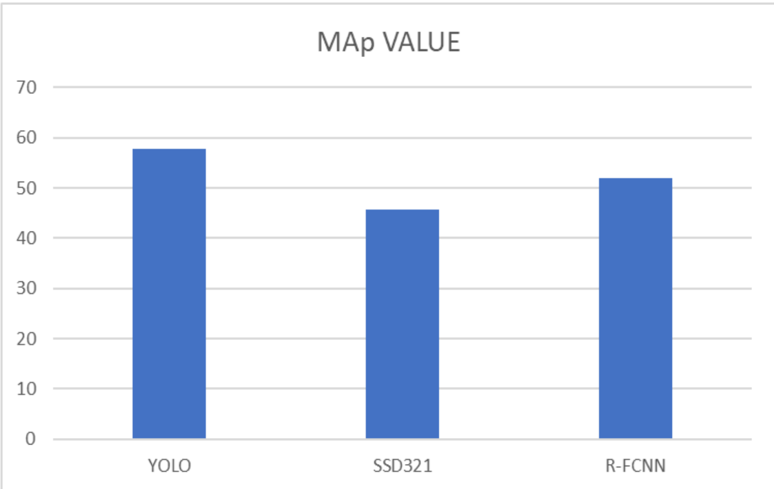

Figure 3. Graphical analysis of MAp value of different Algorithms

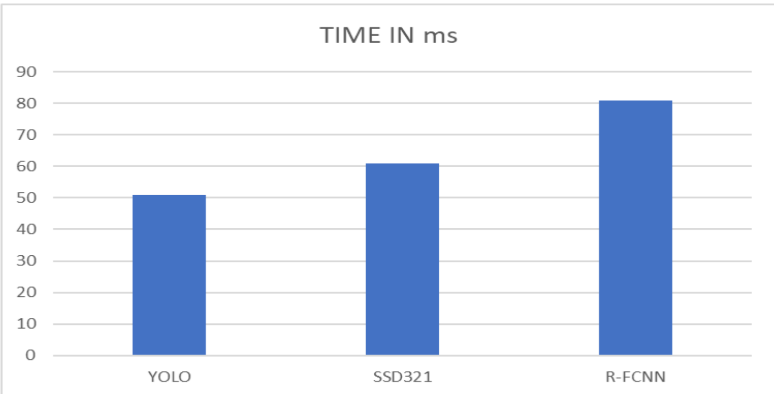

Figure 4. Graphical analysis of Time of different algorithm

ISSN: 2233-7857 IJFGCN

Figure 5. Result

IV. CONCLUSION

In the era of Machine Learning and Artificial Intelligence our project is a benchmark which helps the police officers to stay at their station and still patrol the streets. This will make them to safeguard themselves by only coming up on streets when necessary. Our system is also a great step to bring social distancing in picture till we get a perfect solution towards the global pandemic of COVID-19. Our system (YOLO) is giving great accuracy and also the time required for detection is also less as compared to other detection system such as SSD321 and R-FCNN.

V. FUTURE WORK

In today's world where surveillance is very necessary with the increasing population. Our system will help in making the job of police officers less tedious and still patrol their area in anew efficient way. Our project also brings up a new efficient method in the field of security using Computer Vision.

REFERENCES

11.1. Journal Article

[1] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[2] Ross Girshick. [Fast R-CNN]. In International Conference on Computer Vision (ICCV), 2015.

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. [Fast R-CNN]. In Advances in Neural Information Processing Systems (NIPS), 2015

[4] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, You Only Look Once: Unified, Real- Time Object Detection, arXiv.org, 2016

ISSN: 2233-7857 IJFGCN

[5] Chengji Liu, Yufan Tao, Jiawei Liang, Kai Li, Yihang Chen, Object Detection Based on YOLO Network 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), 2018.

11.2. Websites

[6] https://leonardoaraujosantos.gitbooks.io/artificialinteligence/content/single-shot-etectors

[7] https://www.learnopencv.com/faster-r-cnn-object-detectionwith-pytorch/

[8] https://arxiv.org/pdf/1704.04503.pdf

[9] https://github.com/rbgirshick/fastrcnn/blob/master/lib/utils/nms.py

[10] https://github.com/DocF/Soft-NMS — Python implementation.

[11] https://github.com/DocF/Soft-NMS — Python implementation.

ISSN: 2233-7857 IJFGCN