A computer vision based real-time system for monitoring social distancing in public places

Pratyay Kumar
Department of Information Technology
Indian Institute of Information Technology, Allahabad
pratyay211kumar@gmail.com

***Abstract*—The coronavirus pandemic has spread to over more than 180 countries with around 204 million confirmed cases across the globe at the time of this paper writing (August 11, 2021). Vaccines have not been distributed to a large fraction of the population, thus, practicing social distancing is required to slow down the spread of the virus in our society. This crucial situation motivates us to develop a real-time system for monitoring social distancing practiced between people, especially in public places. Previous social distancing monitoring methods consider a depth sensing camera that provides images with the distance of the objects from the camera, making it convenient to monitor social distancing. In this paper, we consider a mono camera setup that does not provide the depth of objects to the social distancing monitoring system. The You Only Look Once (YOLO) algorithm is used for object detection. We improve the YOLO algorithm by implementing the soft-non-maximum suppression (soft NMS) method over the usual NMS approach for enhanced bounding box prediction in the object detection algorithm. An object tracking algorithm is also implemented to monitor the movement of the pedestrians present in the video data set. The overall monitoring system is demonstrated to be suitable for monitoring social distancing in public places using video surveillance cameras.**

***Index Terms*—COVID-19, Social distancing, You Only Look Once, Non-max suppression, Object detection and tracking, Mono Camera setup.**

I. Introduction

The outbreak of the coronavirus (COVID-19) has changed many lives across the globe. As of April 11, 2021, this virus is known to spread in over 180 countries with 204 million known cases. The world health organization (WHO) declared this situation as a global pandemic. The virus is airborne and can easily spread in our society via droplets released by an infected person. Qualitatively, there are two types of spreading rates of a virus like the COVID-19: fast spreading and slow spreading. A representative plot of the number of cases that can result from both the types of spreading rates is shown in Figure 1. In the fast spreading case, the majority of the population can catch the virus and this situation can overwhelm the hospital capacity, leading to huge loss of lives across the country. By contrast, in the slow spreading case, the number of cases may be manageable by the hospital system capacity, leading to less loss of lives to a noticeable extent compared to the fast spreading case. This less rate of infection can be attained by practicing social distancing to mitigate the spread of the virus.

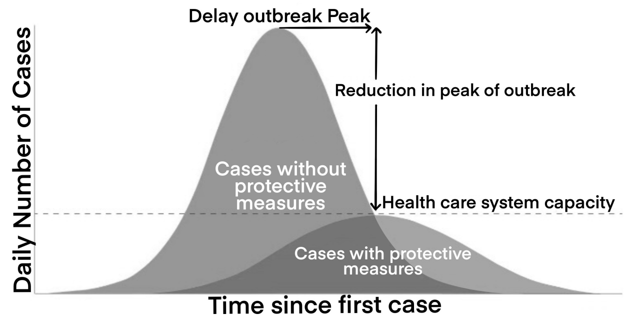

Fig. 1. A diagram depicting the daily number of cases in the fast and slow spreading situations, which correspond to the cases without and with protective measures, respectively.

Practicing social distancing can be challenging in crowded public places such as malls, parks, etc. Therefore, the goal of this paper is to provide a system for social distancing monitoring in such places, which can help local government authorities when patrolling public places during this ongoing pandemic. We describe a novel system for real-time social distancing monitoring currently not proposed in the research literature. This monitoring system works as follows. First, the You Only Look Once (YOLO) algorithm [Liu et al., 2018] for object detection is used to detect pedestrians in an image in real-time. A pre-trained convolutional neural network (CNN) trained over the Microsoft-Common Object in Context (MS-COCO) data set [Lin et al., 2014] is used to predict the bounding boxes in the YOLO algorithm. Typical YOLO implementations use the non-maximum suppression (NMS) approach [Hosang et al., 2017] to predict the bounding boxes of objects in an image. We implement the soft NMS [Bodla et al., 2017] approach to improve the bounding box predictions compared to the NMS method for situations when multiple pedestrians are present in close proximity in an image.

After the bounding box computations in real-time, we compute distances between centroids of each pair of pedestrians and compare that distance against social distancing

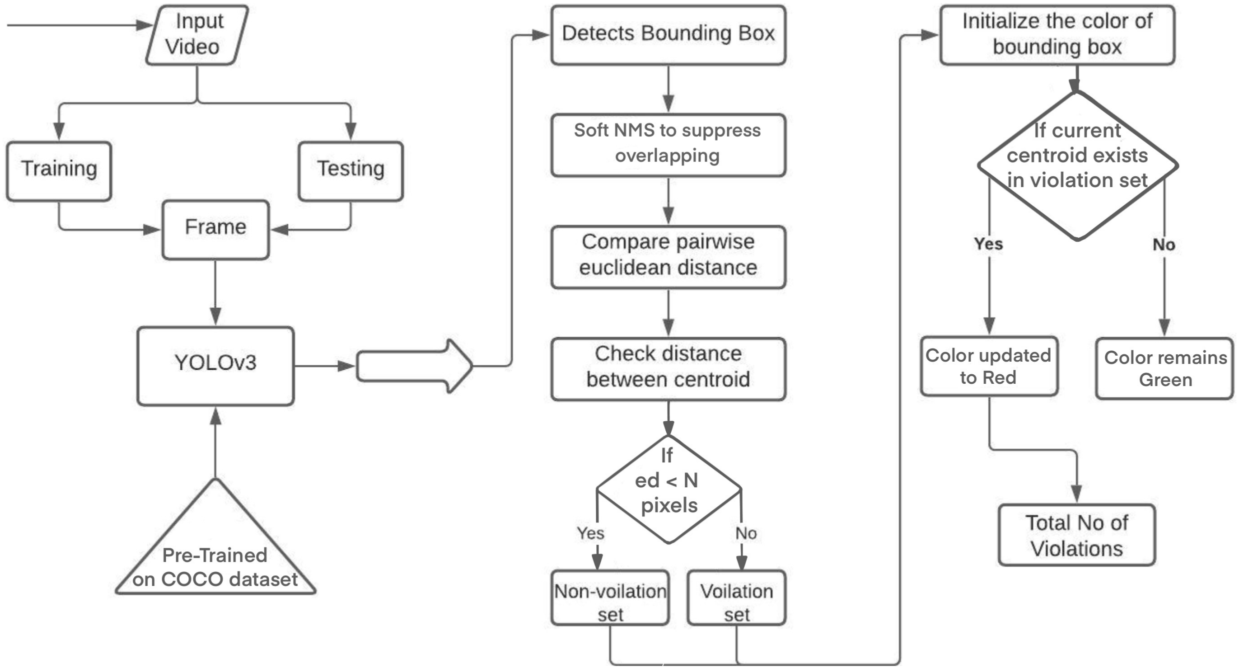

Fig. 2. Flow diagram of the proposed social distancing monitoring framework

guidelines in the monitoring system. We consider a social distancing monitoring setup with monochrome cameras that do not provide the depth of each objects from the camera. To account for this problem, we propose to compute the depth using a modified lens formula [Rosebrock, 2015]. This inclusion of monochrome cameras in the social distancing monitoring system is a novel contribution of this paper. To monitor the movement of all the pedestrians present in the video data set, we implement a real-time object tracking method using a centroid tracking algorithm to keep track of each pedestrian in consecutive image frames. The application and effectiveness of the overall social distancing monitoring framework is demonstrated on a video surveillance data set.

The rest of this paper is structured as follows. In Section II, we discuss related literature on social distancing monitoring systems using machine learning methods. The social distancing monitoring system proposed in this paper is described in detail in Section III. Experimental computational results are presented in Section IV, and we conclude at the end of this paper in Section V.

II. Literature Review

The coronavirus, discovered in late December 2019, popularly known as the COVID-19 has influenced society in unimaginable ways. The number of cases rises everyday as India battles the second wave of the infectious coronavirus. Social distancing is recommended by health experts to slow down the spread of the virus and avoid huge loss of lives as illustrated in Figure 1. Several research works have been carried out since the early 2020 on social distancing monitoring using different machine learning methods and algorithms. This section highlights some of the related works on social distance monitoring approaches available in the literature.

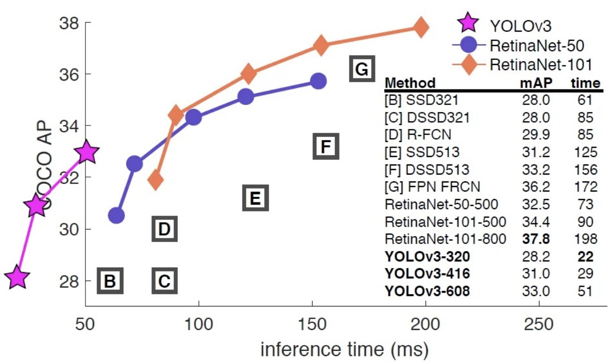

Fig. 3. YOLOv3 runs significantly faster than other detection methods with comparable performance. Figure adopted from Redmon and Farhadi [2018]

Ramadass et al. [2020] proposed a system to use a real-time automated drone for social distancing monitoring. The drone also carries masks and drops the masks to the people in need. Vijay et al. [2021] takes a hardware based approach for social distancing monitoring, and developed a robot that uses the tail tracking principle to continuously queue people and tracks people behavior that violate the social distancing guidelines. Hou et al. [2020], in their model, used the YOLOv3 algorithm for object detection and they transformed the video frames into a top-down view for distance measurement from the camera plane.

The paper Yadav [2020] proposes a framework to use a raspberry Pi4 hardware with a camera to roam in public spaces continuously to forestall the spread of COVID-19. The model

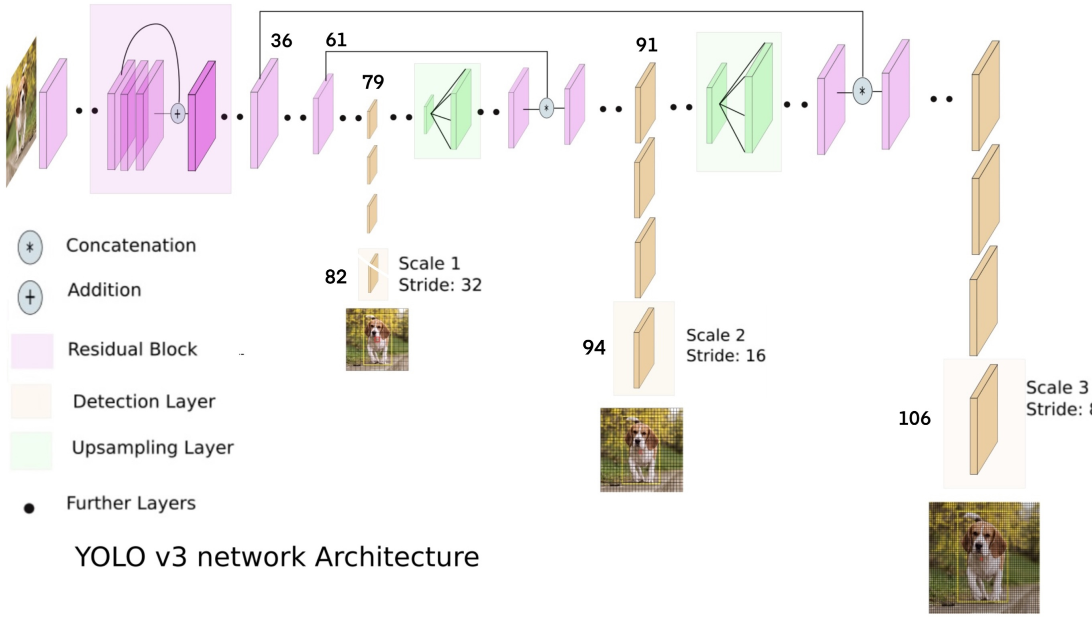

Fig. 4. YOLOv3 Network Architecture. Figure adopted from Kathuria [2018].

was trained on a custom data-set captured using the hardware and camera setup. The camera was installed on the raspberry Pi4 hardware such that it takes continuous recordings like a surveillance camera. A depth sensing camera was used such that social distancing between people can be easily computed using the depth information of objects provided by the camera. Ahmed et al. [2021], in their paper, develop a deep learning platform for monitoring social distancing. The accuracy of the deep learning model was improved by using transfer learning techniques. A computer vision-based social distancing and critical density detection system for COVID-19 was proposed in Yang et al. [2021], in which they provide a deep learning based real-time model for social distancing detection and warning. An audio-visual warning signal is sent to individuals who breach the social distancing guidelines. Cristani et al. [2020] discuss a visual social distancing problem and present a skeleton recognition methodology for monitoring distance between pedestrians. The paper additionally examines the impact of the social setting on individuals on their willingness to follow social separation. Rahim et al. [2021] discusses issues that arise for social distancing monitoring using deep learning under various low light conditions. A single motionless time of flight camera is used in the computations presented in the paper. The proposed framework utilises the YOLOv4 algorithm for object detection. The paper Saponara et al. [2021] discusses the use of YOLOv2 algorithm for social distancing using thermal images of pedestrians.

To summarize, the research literature in the area of social distancing monitoring using machine learning methods is rapidly growing. The practical impact of accurate social distancing monitoring methods is huge and can save lives. The novelty of the work presented in this paper is the inclusion of mono camera setups in the monitoring system and the usage of soft NMS over the standard NMS method for enhanced object detection in crowded public places with the YOLO algorithm.

III. Methodology

In this section, we discuss the proposed social distancing monitoring system. The overall framework and steps involved in the monitoring system are diagrammed in Figure 2, which consists of the following key components: (i) a YOLO algorithm that uses a convolutional neural network to predict the bounding boxes of pedestrians in a surveillance image, (ii) the soft NMS method to obtain the most probable bounding box out of all the bounding boxes predicted by the YOLO algorithm, and (iii) determination of centroids of the bounding boxes corresponding to pedestrians and comparison of distances between the centroids against social distancing guidelines. In addition to the social distancing monitoring system, we implement a real-time object tracking algorithm using a centroid tracking method to monitor the position of each pedestrian. We next describe each of the above components in detail.

A. *You Only Look Once (YOLO)*

We use the well known YOLO algorithm for detecting pedestrians in an image. Several object detection algorithms have been proposed in the literature [Roh and Lee, 2017, Zhan et al., 2007]. In this work, we specifically use the YOLOv3 algorithm due to is known superior performance in the computer vision literature [Redmon and Farhadi, 2018]. Figure 3 shows a qualitative plot of the performance of the YOLOv3 algorithm compared to other popular object detection algorithms, and illustrates that the YOLOv3 algorithm provides a faster inference time with a similar accuracy compared to the other detection algorithms.

All the YOLO algorithms available in the literature use a convolutional neural network (CNN) [Albawi et al., 2017] for object detection. A single neural network is used for the whole image, and the frame is divided into several cells in a grid. The neural network predicts (i) bounding boxes of objects in each cell using some pre-defined anchor boxes, and (ii) probability of containing an object of a particular class in the bounding box. The convolutional network architecture used in the YOLOv3 algorithm is shown in Figure 4. The algorithm uses a larger network to perform feature extraction compared to previous versions with 53 convolution layers and consists residual connections. This larger network capacity increases the accuracy of the algorithm, while maintaining a good inference speed. Other features of the YOLOv3 algorithm that enhance its performance over previous variants are discussed in Redmon and Farhadi [2018]. The output layer of the CNN predicts a class probability (p_c), bounding box coordinates (b_x, b_y), and dimensions (b_h, b_w). These predictions are computed using the following equations:

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= p_w e^{t_w} \\ b_h &= p_h e^{t_h} \end{aligned}$$

in which, t_x, t_y, t_w, and t_h are the bounding box coordinates predicted by the network, c_x and c_y are the cell offset from the top left corner of the image. These bounding box calculations are also illustrated in Figure 5. In this paper, we use a fine-tuned, pre-trained neural network with model weights trained on the COCO dataset obtained from Menon et al. [2021].

B. Soft non-maximum suppression

The YOLO algorithm may predict multiple bounding boxes for a single object. Non-maximum suppression (NMS) [Hosang et al., 2017] can be used to choose the bounding box that has the highest probability of containing the whole object. The neural network in the YOLOv3 algorithm also predicts probabilities of containing an object of different classes in addition to the bounding box coordinates. The NMS method first uses these class probabilities and selects the bounding box that has the highest probability of containing an object. Then, all the other bounding boxes with an intersection over union (IOU) metric of greater than 0.5 with this chosen bounding box are suppressed. The intersection over union (IOU) metric is illustrated in Figure 6, which is computed as the ratio of the area of overlap between the two boxes and the total area occupied by the two boxes. The final output of the NMS step is the bounding box with the highest probability of containing the object, similar to the example illustrated in Figure 8. The NMS approach is an integrated part of the object detection pipeline in the YOLOv3 implementation. Figure 7 shows the pseudo code of the NMS algorithm [Hosang et al., 2017].

The NMS approach of selecting the best bounding box has a crucial issue for situations in which multiple objects of the same class are present in close proximity in an image. Since only one bounding box with the highest probability is chosen, and all the other bounding boxes with very less probability or having the IOU metric greater than 0.5 are suppressed, the approach also suppresses bounding boxes corresponding to different objects of the same class present in close proximity to the current object. As a result, this standard NMS approach has a lower object detection accuracy when multiple objects of the same class are present in close proximity. The soft NMS method described next circumvents this issue and retains the most probable bounding boxes of all the objects of the same class present in close proximity in an image.

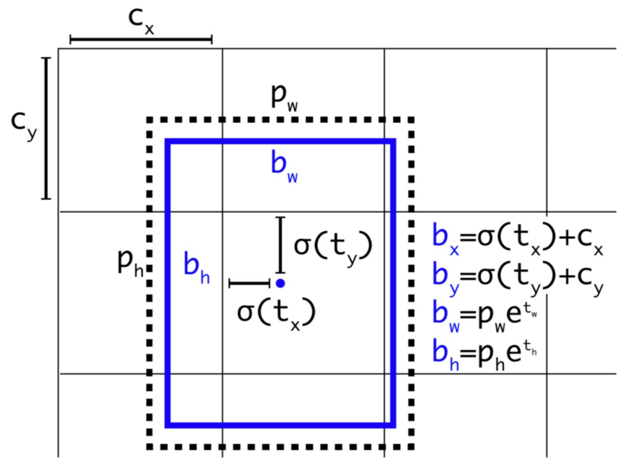

Fig. 5. Bounding box calculations using coordinate predictions. Figure adopted from Redmon and Farhadi [2018].

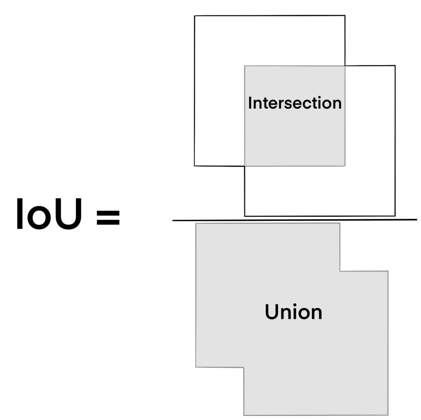

Fig. 6. A schematic depicting the calculation of the intersection over union (IOU) metric between two bounding boxes.

The soft non-maximum (soft NMS) approach to choose the best bounding boxes for object detection was proposed in Bodla et al. [2017]. This method is an improvement over the basic NMS algorithm and works as follows. Rather than suppressing all the bounding boxes that have a probability lower than the highest probability bounding box and an IOU metric greater than 0.5 with the highest probability bounding box, a new probability measure (score) is assigned to those bounding boxes. This score is assigned using an increasing

Algorithm 1 Non-Max Suppression

```
procedure NMS(B,c)
    B_nms ← ∅    Initialise empty set
    for b_i ∈ B do ⇒ Iterate over all boxes
        discard ← False    Take Boolean variable and set it as false. This variable indicates whether b(i) should be kept or discarded.
        for b_j ∈ B do    Start another loop to compare with b(i)
            if same(b_i, b_j) > λ_nms then    If both boxes having same IOU.
                if score(c, b_j) > score(c, b_i) then
                    discard ← True    Compare the scores. If score of b(i) is less than that of b(j), b(i) should be discarded, so set the flag to True.
        if not discard then    Once b(i) is compared with all other boxes and still the discarded flag is False, then b(i) should be considered. So add it to the final list.
            B_nms ← B_nms ∪ b_i
    return B_nms    Do the same procedure for remaining boxes and return the final list.
```

Fig. 7. Non-Max suppression algorithm [Rothe et al., 2014].

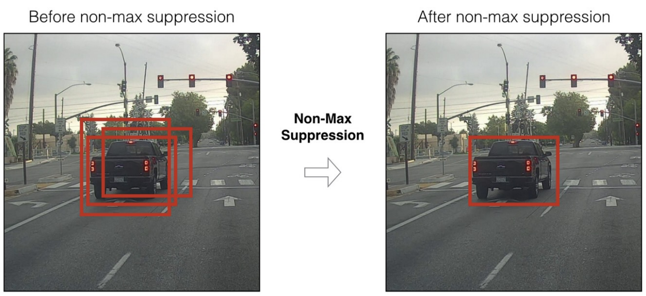

Fig. 8. An example illustrating the non-maximum suppression method. The algorithm picks the bounding box that has the highest probability of containing the whole object out of the three bounding boxes predicted by the neural network.

function ($f(\cdot)$) of the IOU metric such that bounding boxes very close to the highest probability bounding box are assigned a low score. Bounding boxes slightly far from the highest probability bounding box are assigned the same score or the scores are decreased less for those boxes according to the chosen function of the IOU metric ($f(\cdot)$). After this new score assignment step, the next highest score box is selected and the process of assigning new scores to all the bounding boxes is repeated until all the remaining bounding boxes have a low score compared to a chosen threshold.

The pseudo code for the soft NMS method is shown in Figure 10 and the change in code compared to the basic NMS method is highlighted in that figure. The function ($f(\cdot)$) of the IOU metric used to assign the new scores is shown in Figure 9, which is such that the scores of all the boxes that have an IOU metric less than a threshold (N_t) are not changed and the scores of the other boxes with an IOU metric greater than the threshold are decreased linearly. The soft NMS approach described in this subsection is easy to implement and can be integrated in existing YOLOv3 implementations.

$$s_i = \begin{cases} s_i, & \text{iou}(\mathcal{M}, b_i) < N_t \\ s_i(1 - \text{iou}(\mathcal{M}, b_i)), & \text{iou}(\mathcal{M}, b_i) \geq N_t \end{cases},$$

si - score of proposal, bi - box corresponding to proposal i, M - box corresponding to maximum confident, Nt - IOS threshold

Fig. 9. The function used to assign new scores to bounding boxes that have a less probability of containing an object compared to the highest probability bounding box.

C. Distance calculation

As mentioned in Section I, we consider a mono camera setup that does not provide the depth of objects in an image. The spatial coordinates (x, y) of the centroid of each pedestrian in the image is determined from the object detection procedure described above. The depth of the object is required to compute distances between two pedestrians for comparison against social distancing guidelines. We compute this depth using the following lens formula [Rosebrock, 2015]:

$$D = \frac{FW}{P}$$

in which, F is the focal length of the camera, P is the pixel covered by the object, W is the width of the object, and D is the depth of the object from the camera. This computed depth of the object can then be used as the third z coordinate in a 3D coordinate system to compute the relative distance between two pedestrians. Finally, we calculate the following Euclidean distance to compute the distance between two pedestrians:

$$d(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$$

in which, the indices i and j refer to two pedestrians in an image, and $d(i,j)$ is the distance between the two pedestrians.

Input : $\mathcal{B} = \{b_1, .., b_N\}$, $\mathcal{S} = \{s_1, .., s_N\}$, N_t
$\mathcal{B}$ is the list of initial detection boxes
$\mathcal{S}$ contains corresponding detection scores
N_t is the NMS threshold

begin
  $\mathcal{D} \leftarrow \{\}$
  while $\mathcal{B} \neq$ *empty* **do**
    $m \leftarrow$ argmax $\mathcal{S}$
    $\mathcal{M} \leftarrow b_m$
    $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{M}$; $\mathcal{B} \leftarrow \mathcal{B} - \mathcal{M}$
    for b_i *in* $\mathcal{B}$ **do**
      if $iou(\mathcal{M}, b_i) \geq N_t$ **then**
        $\mathcal{B} \leftarrow \mathcal{B} - b_i$; $\mathcal{S} \leftarrow \mathcal{S} - s_i$
      end
      NMS

      $s_i \leftarrow s_i f(iou(\mathcal{M}, b_i))$
      Soft-NMS
    end
  end
  return $\mathcal{D}, \mathcal{S}$
end

Fig. 10. Pseudo code for the soft NMS algorithm for bounding box suppression. The code in the red is replaced with the code in green for the soft NMS method. This algorithm has been adopted from Bodla et al. [2017].

This obtained distance can then be compared against the desired social distancing guidelines for use in the monitoring system.

D. Object tracking

In addition to the social distancing monitoring system using the YOLO and soft-NMS algorithms, we implement a object tracking method to monitor the real-time positions and movement of each pedestrian in the video surveillance data set. This object tracking step complements the social distancing monitoring system to provide a warning sign and information about the pedestrian pairs that are not abiding by social distancing guidelines in consecutive image frames.

A simple centroid tracking approach is implemented. At the beginning of the video, we find the coordinates of the centroid of the bounding box corresponding to each pedestrian in the image and assign a unique identification number (ID) to each pedestrian. In the next frame, we compute the distances between all the pairs of centroids, and the centroids which are closest to each other between two consecutive frames are considered to belong to the same pedestrian. Further, if a new pedestrian enters the surveillance video, a new ID is assigned to that pedestrian. Additionally, if a close centroid pair is not found for a pedestrian with some particular ID for 50 consecutive frames, then that pedestrian is removed from the object tracking database. This object tracking algorithm complements the overall social distancing monitoring system to monitor the movement of each pedestrian in real-time during the video surveillance.

IV. Computation and Results

Implementation

A pre-trained neural network is used for the YOLOv3 implementation. For predictions at the testing phase, a forward propagation pass through the neural network is performed to obtain the bounding boxes surrounding the pedestrians. The network model configurations and weights are taken from `https://pjreddie.com/darknet/yolo`. The soft NMS algorithm for bounding box selection is implemented via the functions available on the OpenCV module. After the bounding box computations, the Euclidean distance between each pedestrian pair is computed and compared against the social distancing guidelines. The pedestrians that violate the distancing guidelines are marked red, and the pedestrians that abide by the guidelines are marked green. The code for the computations presented in this paper is available at https://github.com/Pratyay211/social-distance-monitoring.

Results

The overall social distancing monitoring system is tested on a selected 15 seconds of video data taken from https://github.com/Pratyay211/social-distance-monitoring/blob/main/pedestrians.mp4. A complete video of the performance of the social distancing monitoring system is available at https://github.com/Pratyay211/social-distance-monitoring/blob/main/output.gif.

Figures 11-12 show the output of the monitoring system for some six selected sequential frames out of the 15 second video. We notice in the figures that the monitoring system provides acceptable performance and identifies all the pedestrian pairs that are present in close proximity to each other and marks those pedestrians red. The total number of social distancing violations by the pedestrians is also reported in real-time as shown in the figures. The overall accuracy of the social distancing monitoring system on the video data set is 93.89%. This accuracy was computed by comparing the model predictions to hand labeled social distancing violations on the video data set. The attained accuracy demonstrates the effectiveness of the distancing monitoring system. The soft NMS approach for bounding box selection is particularly effective for object detection in the crowded setting example presented in this paper, and it is a key feature to achieve high performance with the monitoring framework.

Future work

There are several opportunities for improvements in the proposed social distancing monitoring framework. As discussed in Section III, we used a simple modified lens formula to compute the depth of the pedestrians in an image. This approach can be improved by using a higher order and sophisticated depth calculation method to more accurately compute the Euclidean distance between each pedestrian pairs. The convolutional neural network used in the YOLOv3 algorithm can be fine tuned with video surveillance data sets for more accurate object detection on surveillance data sets. Lastly, the monitoring system should be further tested on longer video sets coming from different types of public places to extensively quantify the performance of the system before real-world deployment.

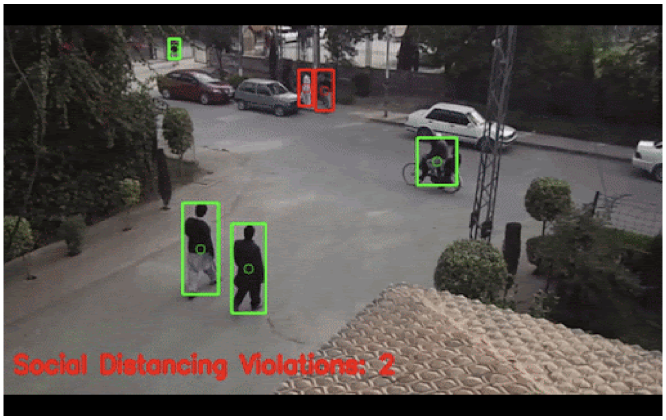

Fig. 11. Prediction of the social distancing monitoring system – Frame 1

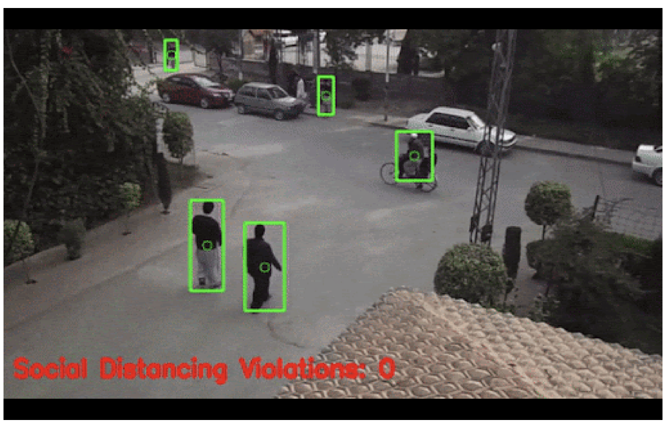

Fig. 12. Prediction of the social distancing monitoring system – Frame 2

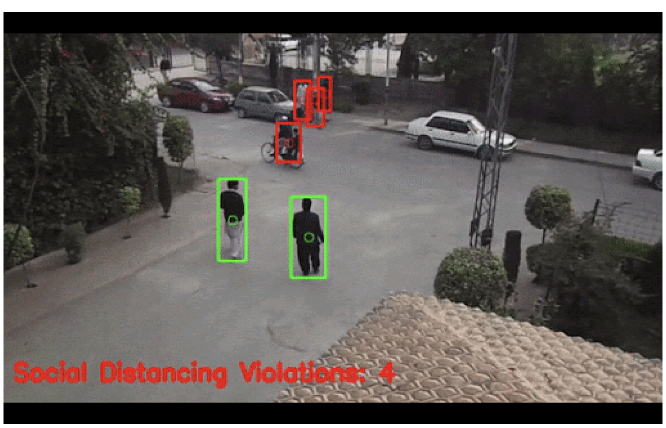

Fig. 13. Prediction of the social distancing monitoring system – Frame 3

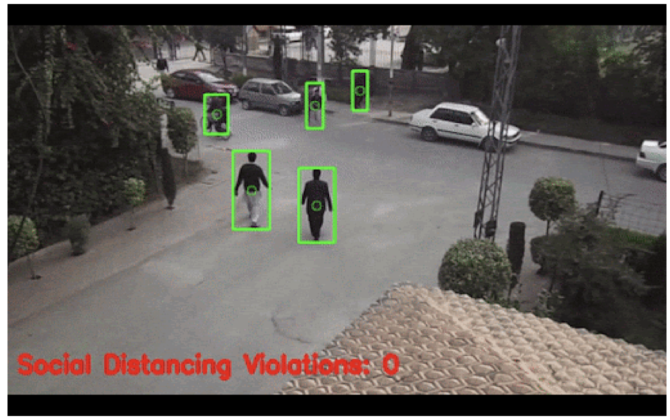

Fig. 14. Prediction of the social distancing monitoring system – Frame 4

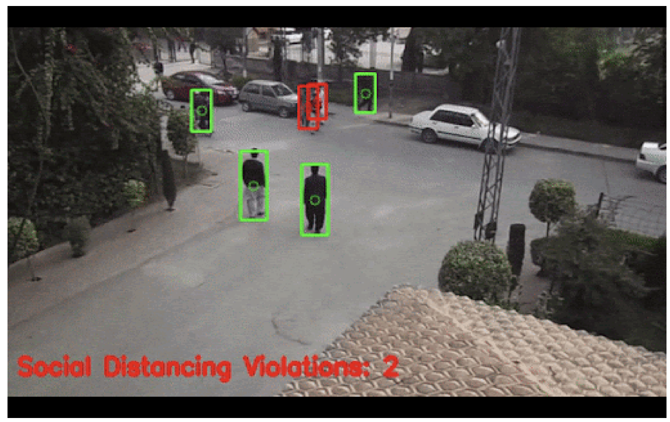

Fig. 15. Prediction of the social distancing monitoring system – Frame 5

V. Conclusions

In this paper, a social distancing monitoring framework using a computer vision method has been presented. The monitoring system uses the YOLOv3 algorithm for object detection, soft NMS for bounding box suppression, and euclidean distance calculation between pedestrian pairs for comparison against social distancing guidelines. This monitoring system is demonstrated to be effective for social distancing monitoring on a 15 second video data set. We consider a mono camera setup that does not provide the depth of objects in an image. This type of camera setup is not considered by other social distancing monitoring systems proposed in the literature. The soft NMS approach for bounding box predictions enhances the object detection accuracy in the crowded public video. Future research may be directed towards improving the accuracy of the social distancing monitoring system and testing on a more wide range of video surveillance data sets. Success of projects in this research area may lead to the practical deployment of social distancing monitoring methods in both private and government public places.

References

I. Ahmed, M. Ahmad, J. J. Rodrigues, G. Jeon, and S. Din. A deep learning-based social distance monitoring framework

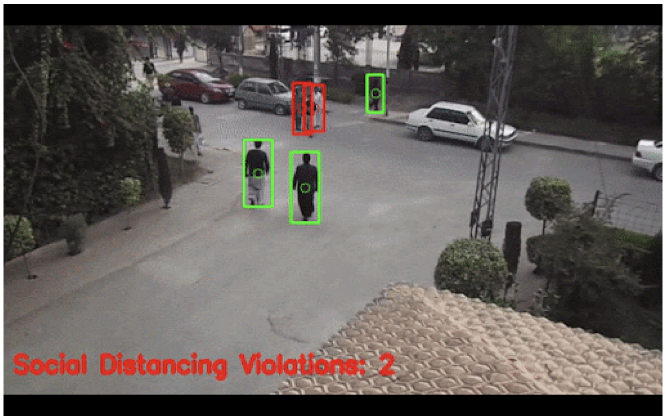

Fig. 16. Our Prediction of the social distancing monitoring system – Frame 6

for covid-19. *Sustainable Cities and Society*, 65:102571, 2021.

S. Albawi, T. A. Mohammed, and S. Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6. Ieee, 2017.

N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017.

M. Cristani, A. Del Bue, V. Murino, F. Setti, and A. Vinciarelli. The visual social distancing problem. *IEEE Access*, 8: 126876–126886, 2020.

J. Hosang, R. Benenson, and B. Schiele. Learning non-maximum suppression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4507–4515, 2017.

Y. C. Hou, M. Z. Baharuddin, S. Yussof, and S. Dzulkifly. Social distancing detection with deep learning model. In *2020 8th International Conference on Information Technology and Multimedia (ICIMU)*, pages 334–338. IEEE, 2020.

A. Kathuria. What's new in yolo v3?, Apr. 2018. URL https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

C. Liu, Y. Tao, J. Liang, K. Li, and Y. Chen. Object detection based on yolo network. In *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pages 799–803. IEEE, 2018.

A. Menon, B. Omman, and S. Asha. Pedestrian counting using yolo v3. In *2021 International Conference on Innovative Trends in Information Technology (ICITIIT)*, pages 1–9. IEEE, 2021.

A. Rahim, A. Maqbool, and T. Rana. Monitoring social distancing under various low light conditions with deep learning and a single motionless time of flight camera. *Plos one*, 16(2):e0247440, 2021.

L. Ramadass, S. Arunachalam, and Z. Sagayasree. Applying deep learning algorithm to maintain social distance in public place through drone technology. *International Journal of Pervasive Computing and Communications*, 2020.

J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

M.-C. Roh and J.-y. Lee. Refining faster-rcnn for accurate object detection. In *2017 fifteenth IAPR international conference on machine vision applications (MVA)*, pages 514–517. IEEE, 2017.

A. Rosebrock. Find distance from camera to object/marker using python and opencv, Jan. 2015. URL https://www.pyimagesearch.com/2015/01/19/find-distance-camera-objectmarker-using-python-opencv/#pyis-cta-modal.

R. Rothe, M. Guillaumin, and L. Van Gool. Non-maximum suppression for object detection by passing messages between windows. In *Asian conference on computer vision*, pages 290–306. Springer, 2014.

S. Saponara, A. Elhanashi, and A. Gagliardi. Implementing a real-time, ai-based, people detection and social distancing measuring system for covid-19. *Journal of Real-Time Image Processing*, pages 1–11, 2021.

A. Vijay, A. Gupta, A. Pal, B. Sriswathi, G. Mathur, and S. K. Alaria. Iot social distancing & monitoring robot for queue. 2021.

S. Yadav. Deep learning based safe social distancing and face mask detection in public areas for covid-19 safety guidelines adherence. *International Journal for Research in Applied Science and Engineering Technology*, 8(7):1368–1375, 2020.

D. Yang, E. Yurtsever, V. Renganathan, K. A. Redmill, and Ü. Özgüner. A vision-based social distancing and critical density detection system for covid-19. *Sensors*, 21(13): 4608, 2021.

C. Zhan, X. Duan, S. Xu, Z. Song, and M. Luo. An improved moving object detection algorithm based on frame difference and edge detection. In *Fourth international conference on image and graphics (ICIG 2007)*, pages 519–523. IEEE, 2007.