SCTE•ISBE CABLE-TEC EXPO® 2020
VIRTUAL EXPERIENCE » OCTOBER 12-15

2020 Fall Technical Forum
SCTE·ISBE • NCTA • CABLELABS®

Enforcing Social Distancing Using Computer Vision and Deep Learning

A Technical Paper prepared for SCTE•ISBE by

Wael Guibene
Director – Wireless R&D
Charter Communications
6360 S Fiddlers Green Cir, Englewood, CO 80111
Wael.Guibene@charter.com

Hossam Hmimy
Sr. Director – Wireless R&D
Charter Communications
6360 S Fiddlers Green Cir, Englewood, CO 80111
Hossam.Hmimy@charter.com

Table of Contents

List of Figures

1. Introduction

A defining moment of the century, so far, is the unprecedented impact that COVID-19 has brought to the economies world-wide, the populations and defining new norms in society.

Our paper details how we can enforce the new rules of society like social distancing and wearing face masks in open-spaces using computer vision and deep learning through:

- Detecting people on a particular scene,
- Calculating and monitoring the distances between the different people,
- Tracking movements and segregating moving people (might come close to each other during brief moments) from people standing still and violating the social distancing rules, and
- Creating alerts (audio, visual, light…) to enforce social distancing.

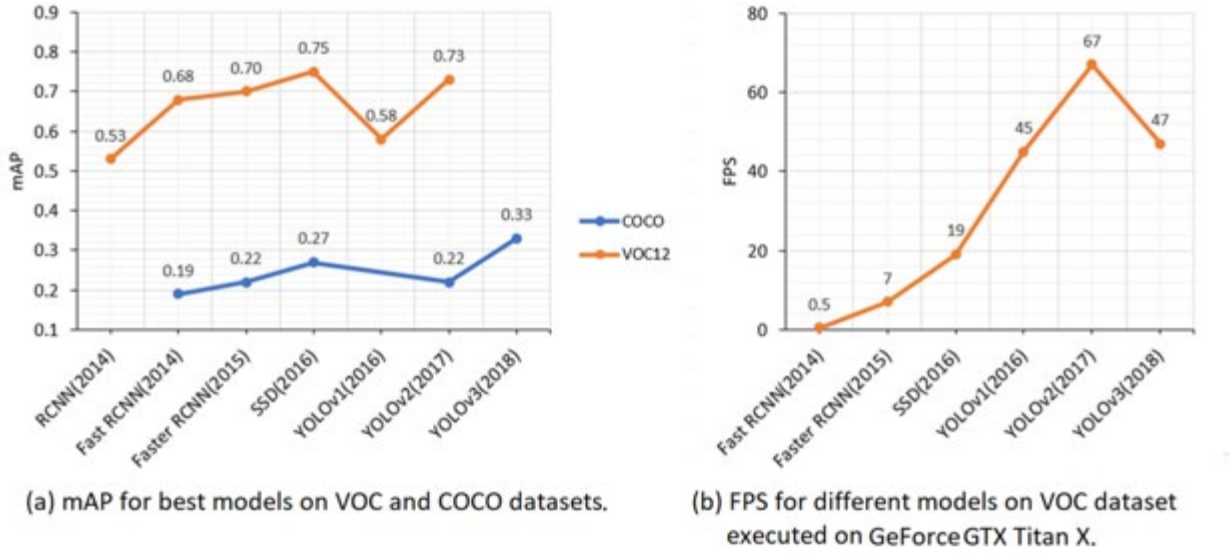
Our approach also ensures that, in confined spaces, we count people and create visual and audio alerts when the number of people exceeds the Center for Disease Control (CDC) guidelines (10 people per room).

We detail in the paper the computer vision and deep learning frameworks we used to achieve high confidence in detecting human presence, calculating and calibrating distances in the frames, and removing false positives (eg. people crossing paths while walking versus people standing still).

2. Computer Vision and DL-based Social Distancing Application

The emergence of deep learning has brought the best performing techniques for a wide variety of tasks and challenges including medical diagnosis, machine translation, speech recognition, and a lot more. Most of these tasks are centered around object classification, detection, segmentation, tracking, and recognition.

In recent years, the convolution neural network (CNN) based architectures have shown significant performance improvements that are leading towards the high quality of object detection, as shown in Fig. 1, which presents the performance of such models in terms of mAP and FPS on standard benchmark datasets, PASCAL-VOC and MS-COCO, and similar hardware resources. In this paper, a deep learning-based framework is proposed that utilizes object detection and tracking models to aid in the social distancing remedy for dealing with the escalation of COVID-19 cases. To maintain the balance of speed and accuracy, YOLO v3 alongside Centroid tracking are utilized as object detection and tracking approaches while surrounding each detected object with bounding boxes. Later, these bounding boxes are utilized to compute the pairwise L2 norm with computationally efficient vectorized representation for identifying the clusters of people not obeying the order of social distancing. Furthermore, to visualize the clusters in the live stream, each bounding box is color-coded based on its association with the group where people belonging to the same group are represented with the same color. Each surveillance frame is also accompanied with the streamline plot depicting the statistical count of the number of social groups and an index term (violation index) representing the ratio of the number of people to the number of groups. Furthermore, estimated violations can be computed by multiplying the violation index with the total number of social groups.



(a) mAP for best models on VOC and COCO datasets.

(b) FPS for different models on VOC dataset executed on GeForce GTX Titan X.

Figure 1 Performance overview of the most popular object detection models on PASCAL-VOC and MS-COCO datasets

3. System Description

In this section, we descibe the end-to-end (E2E) system flow for people detection, tracking and social distance measurement.

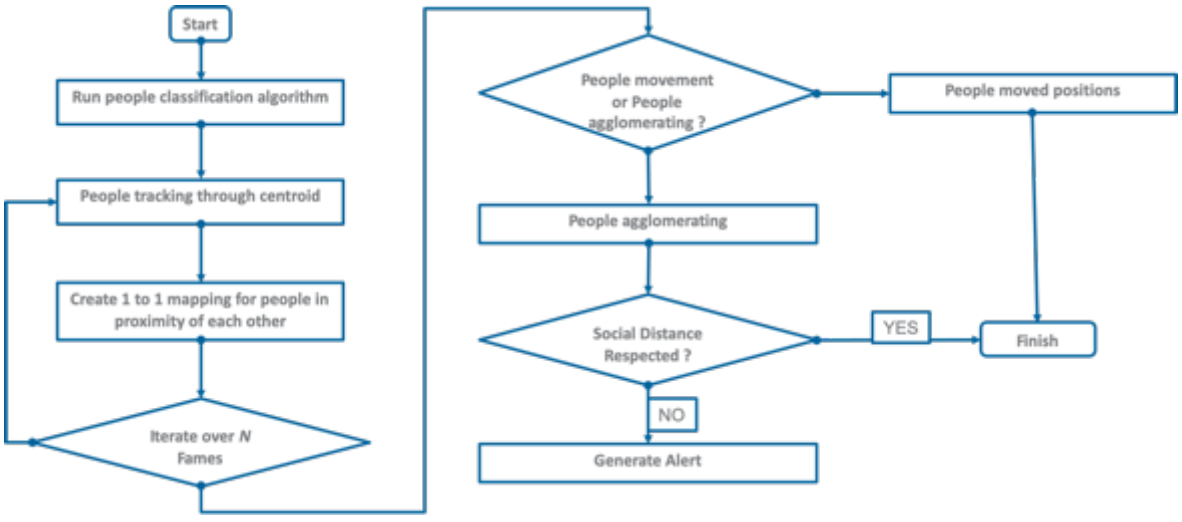


Figure 2. E2E Detection and Tracking Flow.

The flow as depicted in Fig.2 starts when the system is receiving frames via RTSP from a networked camera (WiFi, Ethernet, CBRS, LTE…). The algorithm performs the object classification until classifying objects as people. Each person in the scene is tracked via centroid algorithm. A "security/privacy zone" surrounding each person of X ft is created and the algorithms keeps tracking distances between the closet centroids to each other. In order to minimize false alarm, we re-iterate over N frames in order to rule out people moving from people standing still. If the same centroids are identified in a single group over the N frames, the algorithm classifies the group as standing still and conglomerating, if the group is not respecting the social distance of X an alert (visual, light, voice) is issued to remind the group of social distancing rules.

4. People Detection Algorithm

From the Camera stream, we run an object identification and classification algorithm to infer with high precision the presence of people in the video stream.

Each frame is sub-divided into smaller Regions of Interests (ROIs). Each boundary box or ROI contains 5 elements: (x, y, w, h) and a box confidence score. The confidence score reflects how likely the box contains an object (objectness) and how accurate is the boundary box. We normalize the bounding box width w and height h by the image width and height. x and y are offsets to the corresponding cell. Hence, x, y, w and h are all between 0 and 1. Each cell has 20 conditional class probabilities. The conditional class probability is the probability that the detected object belongs to a particular class (one probability per category for each cell). So, our approach's prediction has a shape of (S, S, B×5 + C) = (7, 7, 2×5 + 20) = (7, 7, 30). The major concept is to build a CNN to predict a (7, 7, 30) tensor. It uses a CNN to reduce the spatial dimension to 7×7 with 1024 output channels at each location. The algorithm performs a linear regression using two fully connected layers to make 7×7×2 boundary box predictions. To make a final prediction, we keep those with high box confidence scores (greater than 25%) as our final predictions.

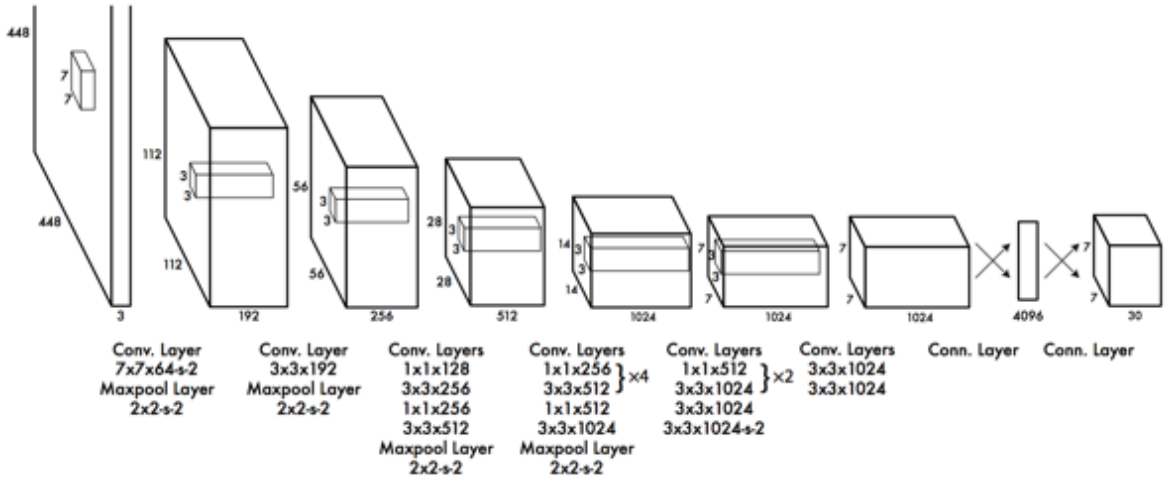


Figure 3 Classification using YOLO v3 Framework.

5. People Tracking via Centroid

Step 1: Accept bounding box coordinates and compute centroids

The centroid tracking algorithm assumes that we are passing in a set of bounding box (x,y)-coordinates for each detected object in every single frame. These bounding boxes can be produced by our object detector provided that they are computed for every frame in the video. Once we have the bounding box coordinates we must compute the "centroid", or more simply, the center (x,y)-coordinates of the bounding box.

Figure 4 bellow demonstrates accepting a set of bounding box coordinates and computing the centroid. Since these are the first initial set of bounding boxes presented to our algorithm we will assign them unique IDs.

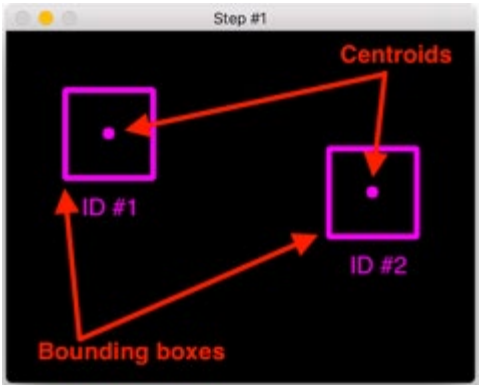


Figure 4. Centroid tracking step 1.

Step 2: Compute Euclidean distance between new bounding boxes and existing objects.

For every subsequent frame in our video stream we apply Step #1 of computing object centroids; however, instead of assigning a new unique ID to each detected object (which would defeat the purpose of object tracking), we first need to determine if we can associate the new object centroids (yellow) with the old object centroids (purple). To accomplish this process, we compute the Euclidean distance (highlighted with green arrows) between each pair of existing object centroids and input object centroids. From figure 5, we can see that we have this time detected three objects in our image. The two pairs that are close together are two existing objects. We then compute the Euclidean distances between each pair of original centroids (yellow) and new centroids (purple). But how do we use the Euclidean distances between these points to actually match them and associate them? → Step 3

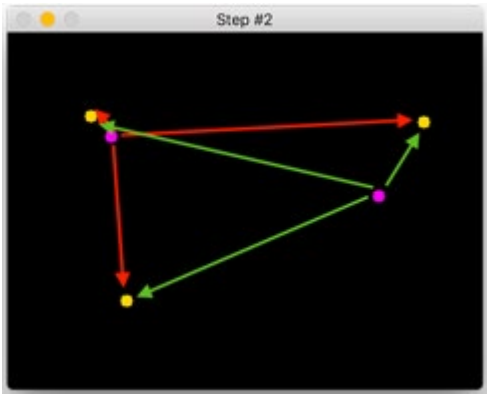


Figure 5. Centroid tracking step 2.

Step 3: Update (x, y)-coordinates of existing objects

The primary assumption of the centroid tracking algorithm is that a given object will potentially move between subsequent frames, but the distance between the centroids for frames F_t and F_(t+1) will be smaller than all other distances between objects. Therefore, if we choose to associate centroids with minimum distances between subsequent frames we can build our object tracker. In figure below, we can see how our centroid tracker algorithm chooses to associate centroids that minimize their respective Euclidean distances.

The new point that appears at the bottom left mean that we see a new object we need to register → Step 4

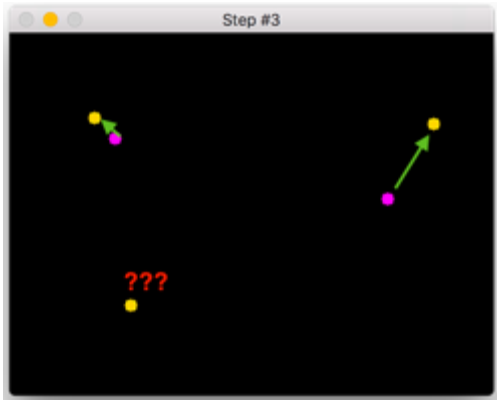


Figure 6. Centroid tracking step 3.

Step 4: Register new objects

In the event that there are more input detections than existing objects being tracked, we need to register the new object. "Registering" simply means that we are adding the new object to our list of tracked objects by:

- Assigning it a new object ID, and
- Storing the centroid of the bounding box coordinates for that object, then
- We can then go back to Step #2 and repeat the pipeline of steps for every frame in our video stream.

Figure 7 depicts the process of using the minimum Euclidean distances to associate existing object IDs and then registering a new object.

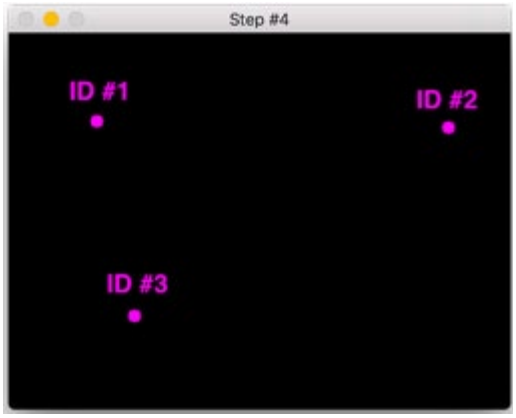


Figure 7 Centroid tracking step 4

Step 5: Deregister old objects

We deregister old objects when they cannot be matched to any existing objects for a total of N subsequent frames.

6. Social Distancing PoC

In order to validate our approach, we implemented and validated our algorithms in our offices in Greenwood Village.

Bellow figures depict the outcome of the PoC:

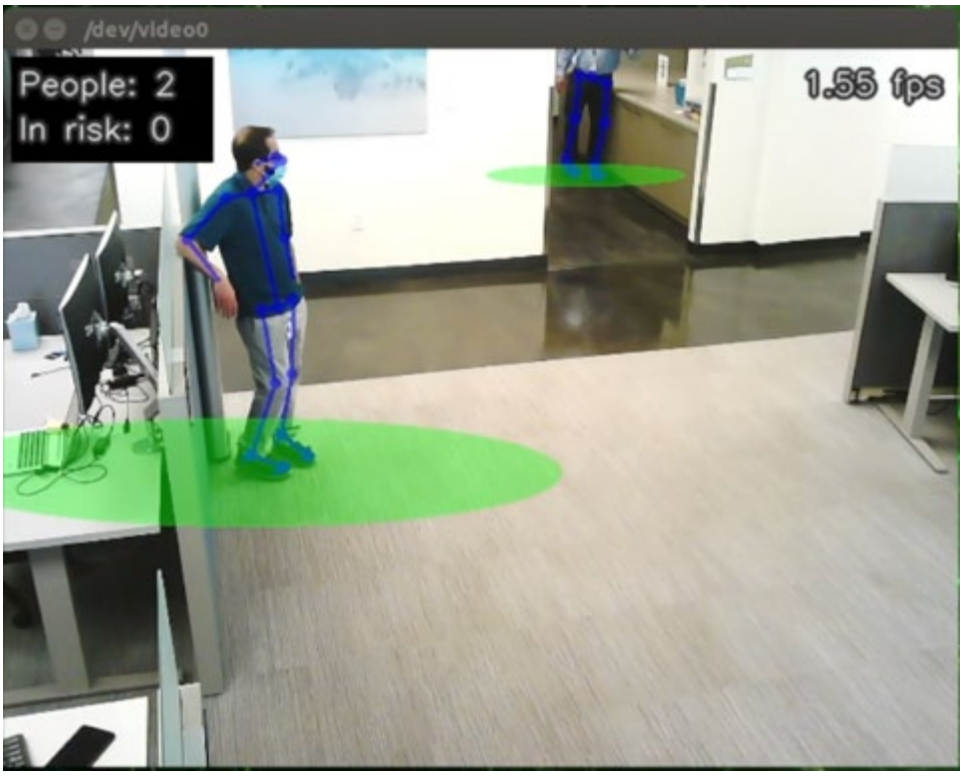


Figure 8. Social Distance fully respected.

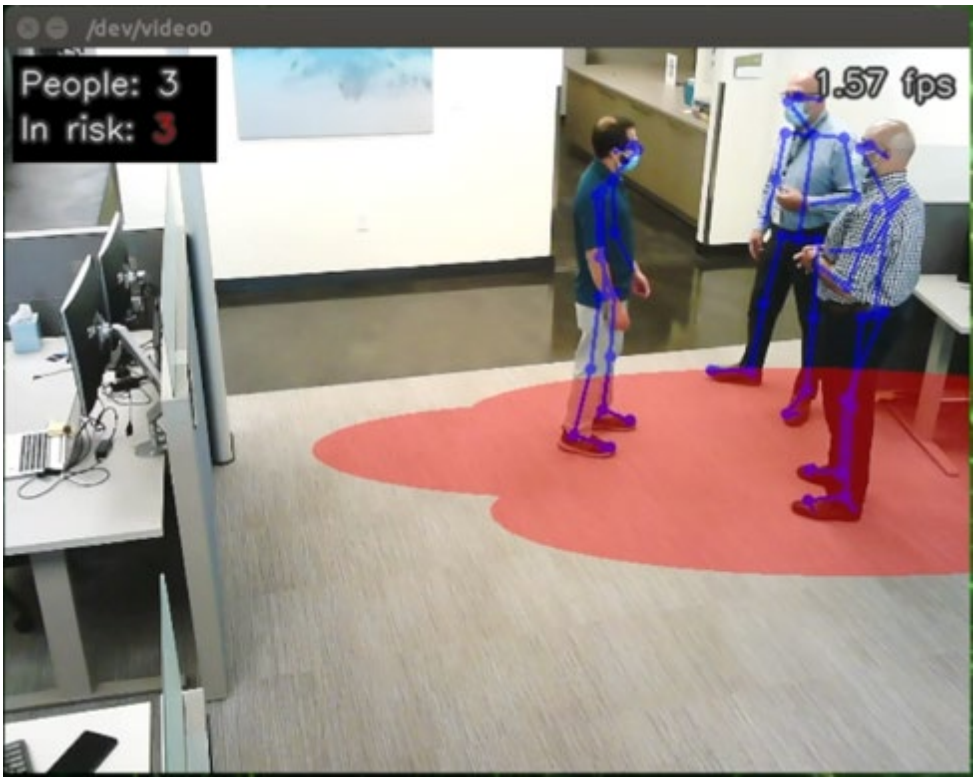


Figure 9. Social Distance not respected.

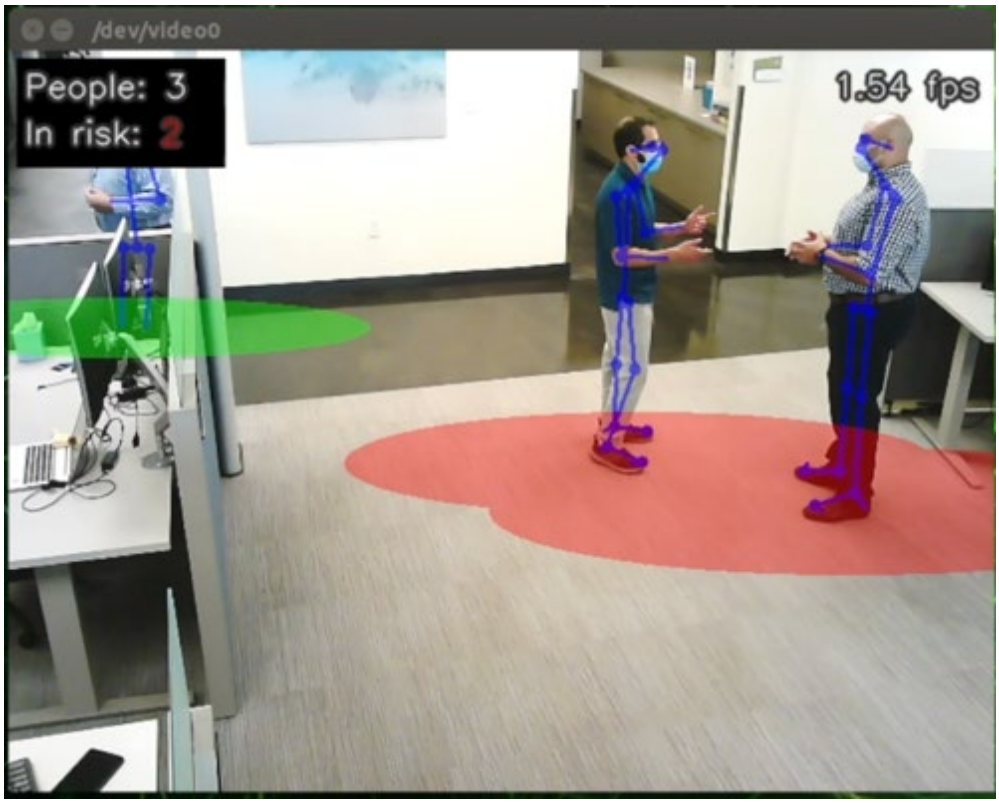


Figure 10. Social Distance partially respected.

7. Conclusion

In this paper, we presented a novel approach to an AI-assisted social distancing application for safer back to work situations.

This solution can be applied to indoor or outdoor scenarios ensuring social distances are met and respected.

We have deployed and tested our solution in our lab, and it has shown promising results and good accuracy for measuring distances and alerting (via sound and lights) when social distancing is not respected.

Bibliography & References

[1] S. A. Niyogi and E. H. Adelson, "Analyzing gait with spatiotemporal surfaces," in Proceedings of 1994 IEEE Workshop on Motion of Nonrigid and Articulated Objects. IEEE, 1994, pp. 64–69.

[2] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," IEEE transactions on neural networks and learning systems, vol. 30, no. 11, pp. 3212–3232, 2019.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in Advances in neural information processing systems, 2015, pp. 91–99.

[5] X. Chen and A. Gupta, "An implementation of faster rcnn with study for region sampling," arXiv preprint arXiv:1702.02138, 2017.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE 10 conference on computer vision and pattern recognition, 2016, pp. 779– 788.

[7] M. Putra, Z. Yussof, K. Lim, and S. Salim, "Convolutional neural network for person and car detection using yolo framework," Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. 10, no. 1-7, pp. 67–71, 2018.

[8] R. Eshel and Y. Moses, "Homography based multiple camera detection and tracking of people in a dense crowd," in 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2008, pp. 1–8.

[9] D.-Y. Chen, C.-W. Su, Y.-C. Zeng, S.-W. Sun, W.-R. Lai, and H.- Y. M. Liao, "An online people counting system for electronic advertising machines," in 2009 IEEE International Conference on Multimedia and Expo. IEEE, 2009, pp. 1262–1265.

[10] C.-W. Su, H.-Y. M. Liao, and H.-R. Tyan, "A vision-based people counting approach based on the symmetry measure," in 2009 IEEE International Symposium on Circuits and Systems. IEEE, 2009, pp. 2617–2620.

[11] J. Yao and J.-M. Odobez, "Fast human detection from joint appearance and foreground feature subset covariances," Computer Vision and Image Understanding, vol. 115, no. 10, pp. 1414–1426, 2011.

[12] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors," International Journal of Computer Vision, vol. 75, no. 2, pp. 247–266, 2007.

