A Vision-based Social Distancing and Critical Density Detection System for COVID-19

Dongfang Yang∗ Ekim Yurtsever∗ Vishnu Renganathan
Keith A. Redmill Ümit Özgüner
The Ohio State University, Columbus, OH 43210, USA
{yang.3455,yurtsever.2, renganathan.5, redmill.1, ozguner.1}@osu.edu

Abstract

Social distancing has been proven as an effective measure against the spread of the infectious COroNaVirus Disease 2019 (COVID-19). However, individuals are not used to tracking the required 6-feet (2-meters) distance between themselves and their surroundings. An active surveillance system capable of detecting distances between individuals and warning them can slow down the spread of the deadly disease. Furthermore, measuring social density in a region of interest (ROI) and modulating inflow can decrease social distancing violation occurrence chance.

On the other hand, recording data and labeling individuals who do not follow the measures will breach individuals’ rights in free-societies. Here we propose an Artificial Intelligence (AI) based real-time social distancing detection and warning system considering four important ethical factors: (1) the system should never record/cache data, (2) the warnings should not target the individuals, (3) no human supervisor should be in the detection/warning loop, and (4) the code should be open-source and accessible to the public. Against this backdrop, we propose using a monocular camera and deep learning-based real-time object detectors to measure social distancing. If a violation is detected, a non-intrusive audio-visual warning signal is emitted without targeting the individual who breached the social distancing measure. Also, if the social density is over a critical value, the system sends a control signal to modulate inflow into the ROI. We tested the proposed method across real-world datasets to measure its generality and performance. The proposed method is ready for deployment, and our code is open-sourced

1. Introduction

Social distancing is an effective measure against the novel COroNaVirus Disease 2019 (COVID-19) pandemic. However, the general public is not used to keep an imaginary safety bubble around themselves. An automatic warning system can help and augment the perceptive capabilities of individuals.

Deploying such an active surveillance system requires serious ethical considerations and smart system design. The first challenge is privacy. If data is recorded and stored, the privacy of individuals may be violated intentionally or unintentionally. As such, the system must be real-time without any data storing capabilities.

Second, the detector must not discriminate. The safest way to achieve this is by building an AI-based detection system. Removing the human out of the detection loop may not be enough – the detector must also be design free. Domain-specific systems with hand-crafted feature extractors may lead to malign designs. A connectionist machine learning system, such as a deep neural network without any feature-based input space, is much fairer in this sense, with one caveat: the distribution of the training data must be fair.

Another critical aspect is being non-intrusive. Individuals should not be targeted directly by the warning system. A non-alarming audio-visual cue can be sent to the vicinity of the social distancing breach to this end.

The system must be open-sourced. This is crucial for establishing trust between the active surveillance system and society.

Against this backdrop, we propose a non-intrusive augmentative AI-based active surveillance system for sending omnidirectional visual/audio cues when a social distancing breach is detected. The proposed system uses a pre-trained deep convolutional neural network (CNN) to detect individuals with bounding boxes in a given monocular camera frame. Then, detections in the image domain are transformed into real-world bird’s-eye view coordinates. If
a distance smaller than the threshold is detected, the system emits a non-alarming audio-visual cue. Simultaneously, the system measures social density. If the social density is over a critical threshold, the system sends an advisory inflow modulation signal to prevent overcrowding.

Our main contributions are:

- A novel vision-based real-time social distancing and critical social density detection system
- Definition of critical social density and a statistical approach to measuring it
- Measurements of social distancing and critical density statistics of common crowded places such as the New York Central Station, an indoor mall, and a busy town center in Oxford.

2. Related Work

Social distancing for COVID-19. COVID-19 has caused severe acute respiratory syndromes around the world since December 2019 [11]. Recent work showed that social distancing is an effective measure to slow down the spread of COVID-19 [1]. Social distancing is defined as keeping a minimum of 2 meters (6 feet) apart from each individual to avoid possible contact. Further analysis [12] also suggests that social distancing has substantial economic benefits. COVID-19 may not be completely eliminated in the short term, but an automated system that can help monitoring and analyzing social distancing measures can greatly benefit our society.

Pedestrian detection. Pedestrian detection can be regarded as either a part of a general object detection problem or as a specific task of detecting pedestrians only. A detailed survey of 2D object detectors, as well as datasets, metrics, and fundamentals, can be found in [13]. Another survey [14] focuses on deep learning approaches for both generic object detection and pedestrian detection. State-of-the-art object detectors use deep learning approaches, which are usually divided into two categories. The first one is called two-stage detectors, mostly based on R-CNN [15], which starts with region proposals and then performs the classification and bounding box regression. The second one is called one-stage detectors, of which the famous models are YOLOv1-v4 [16, 19, 20, 10], SSD [21], RetinaNet [22], and EfficientDet [23]. In addition to these anchor-based approaches, there are also some anchor-free detectors: CornerNet [24], CenterNet [25], FCOS [26], and RepPoints [27]. These models were usually evaluated on datasets of Pascal VOC [28, 29] and MS COCO [30]. The accuracy and real-time performance of these approaches are good enough for deploying pre-trained models for social distancing detection.

Social distancing monitoring. Emerging technologies can assist in the practice of social distancing. A recent work [2] has identified how emerging technologies like wireless, networking, and artificial intelligence (AI) can enable or even enforce social distancing. The work discussed possible basic concepts, measurements, models, and practical scenarios for social distancing. Another work [3] has classified various emerging techniques as either human-centric or smart-space categories, along with the SWOT analysis of the discussed techniques. A specific social distancing monitoring approach [4] that utilizes YOLOv3 and Deepsort was proposed to detect and track pedestrians followed by calculating a violation index for non-social-distancing behaviors. The approach is interesting but results do not contain any statistical analysis. Furthermore, there is no implementation or privacy-related discussion other than the violation index. Social distancing monitoring is also defined as a visual social distancing (VSD) problem in [5]. The work introduced a skeleton detection based approach for inter-personal distance measuring. It also discussed the effect of social context on people’s social distancing and raised the concern of privacy. The discussions are inspirational but again it does not generate solid results for social distancing monitoring and leaves the question open.

Very recently, several prototypes utilizing machine learning and sensing technologies have been developed to help social distancing monitoring. Landing AI [31] has proposed a social distancing detector using a surveillance camera to highlight people whose physical distance is below the
4.1. Problem formulation

We define a scene at time \( t \) as a 6-tuple \( S = (\textbf{I}, A_0, d_c, c_1, c_2, U_0) \), where \( \textbf{I} \in \mathbb{R}^{H \times W \times 3} \) is an RGB image captured from a fixed monocular camera with height \( H \) and width \( W \). \( A_0 \in \mathbb{R} \) is the area of the ROI on the ground plane in real world and \( d_c \in \mathbb{R} \) is the required minimum physical distance. \( c_1 \) is a binary control signal for sending a non-intrusive audio-visual cue if any inter-pedestrian distance is less than \( d_c \). \( c_2 \) is another binary control signal for controlling the entrance to the ROI to prevent overcrowding. Overcrowding is detected with our novel definition of critical social density \( \rho_c \). \( \rho_c \) ensures social distancing violation occurrence probability stays lower than \( U_0 \). \( U_0 \) is an empirically decided threshold such as 0.05.

Problem 1. Given \( S \), we are interested in finding a list of pedestrian pose vectors \( P = (\textbf{p}_1, \textbf{p}_2, \cdots, \textbf{p}_n) \), \( \textbf{p} \in \mathbb{R}^2 \), in real-world coordinates on the ground plane and a corresponding list of inter-pedestrian distances \( D = (d_{1,2}, \cdots, d_{1,n}, d_{2,3}, \cdots, d_{2,n}, \cdots, d_{n-1,n}) \), \( d \in \mathbb{R}^+ \). \( n \) is the number of pedestrians in the ROI. Also, we are interested in finding a critical social density value \( \rho_c \). \( \rho_c \) should ensure the probability \( p(d > d_c | \rho < \rho_c) \) stays over \( 1 - U_0 \), where we define social density as \( \rho := n/A_0 \).

Once Problem 1 is solved, the following control algorithm can be used to warn/advising the population in the ROI.

Algorithm 1. If \( d \leq d_c \), then a non-intrusive audio-visual cue is activated with setting the control signal \( c_1 = 1 \), otherwise \( c_1 = 0 \). In addition, If \( \rho > \rho_c \), then entering the area is not advised with setting \( c_2 = 1 \), otherwise \( c_2 = 0 \).

Our solution to Problem 1 starts below.

4.2. Pedestrian detection in the image domain

First, pedestrians are detected in the image domain with a deep CNN model trained on a real-world dataset:

\[ \{T_i\}_k = f_{\text{cnn}}(\textbf{I}) \]  

\( f_{\text{cnn}} : \textbf{I} \rightarrow \{T_i\}_n \) maps an image \( \textbf{I} \) into \( n \) tuples \( T_i = (l_i, \textbf{b}_i, s_i) \), \( \forall i \in \{1, 2, \cdots, n\} \). \( n \) is the number of detected objects. \( l_i \in L \) is the object class label, where \( L \), the set of object labels, is defined in \( f_{\text{cnn}} \). \( \textbf{b}_i = (\textbf{b}_{i,1}, \textbf{b}_{i,2}, \textbf{b}_{i,3}, \textbf{b}_{i,4}) \) is the associated bounding box (BB) with four corners. \( \textbf{b}_{i,j} = (x_{i,j}, y_{i,j}) \) gives pixel indices in the image domain. The second sub-index \( j \) indicates the corners at top-left, top-right, bottom-left, and bottom-right respectively. \( s_i \) is the corresponding detection score. Implementation details of \( f_{\text{cnn}} \) is given in Section 5.1.

We are only interested in the case of \( l = \text{‘person’} \). We define \( \textbf{p}_i \), the pixel pose vector of person \( i \), with using the middle point of the bottom edge of the BB:

\[ \textbf{p}_i := \frac{(\textbf{b}_{i,3} + \textbf{b}_{i,4})}{2} \]  

4.3. Image to real-world mapping

The next step is obtaining the second mapping function \( h : \textbf{p} \rightarrow \textbf{p} \). \( h \) is an inverse perspective transformation...
function that maps \( p' \) in image coordinates to \( p \in \mathbb{R}^2 \) in real-world coordinates. \( p \) is in 2D bird’s-eye-view (BEV) coordinates by assuming the ground plane \( z = 0 \). We use the following well-known inverse homography transformation \(^{[38]}\) for this task:

\[
\mathbf{p}^{\text{bev}} = \mathbf{M}^{-1} \mathbf{p}^{\text{im}},
\]

(3)

where \( \mathbf{M} \in \mathbb{R}^{3 \times 3} \) is a transformation matrix describing the rotation and translation from world coordinates to image coordinates. \( \mathbf{p}^{\text{im}} = [x', y', 1]^T \) is the homogeneous representation of \( p' = [x', y'] \) in image coordinates, and \( \mathbf{p}^{\text{bev}} = [x^{\text{bev}}, y^{\text{bev}}, 1]^T \) is the homogeneous representation of the mapped pose vector.

The world pose vector \( p \) is derived from \( \mathbf{p}^{\text{bev}} \) with \( p = [x^{\text{bev}}, y^{\text{bev}}] \).

4.4. Social distancing detection

After getting \( P = (p_1, p_2, \ldots, p_n) \) in real-world coordinates, obtaining the corresponding list of inter-pedestrian distances \( D \) is straightforward. The distance \( d_{i,j} \) for pedestrians \( i \) and \( j \) is obtained by taking the Euclidean distance between their pose vectors:

\[
d_{i,j} = \|p_i - p_j\|.
\]

(4)

And the total number of social distancing violations \( v \) in a scene can be calculated by:

\[
v = \sum_{i=1}^{n} \sum_{j=1 \atop j \neq i}^{n} \mathbb{I}(d_{i,j}),
\]

(5)

where \( \mathbb{I}(d_{i,j}) = 1 \) if \( d_{i,j} < d_c \), otherwise 0.

4.5. Critical social density estimation

Finally, we want to find a critical social density value \( \rho_c \) that can ensure the social distancing violation occurrence probability stays below \( U_0 \). It should be noted that a trivial solution of \( \rho_c = 0 \) will ensure \( v = 0 \), but it has no practical use. Instead, we want to find the maximum critical social density \( \rho_c \) that can still be considered safe.

To find \( \rho_c \), we propose to conduct a simple linear regression using social density \( \rho \) as the independent variable and the total number of violations \( v \) as the dependent variable:

\[
v = \beta_0 + \beta_1 \rho + \epsilon,
\]

(6)

where \( \beta = [\beta_0, \beta_1] \) is the regression parameter vector and \( \epsilon \) is the error term which is assumed to be normal. The regression model is fitted with the ordinary least squares method. Fitting this model requires training data. However, once the model is learned, data is not required anymore. After deployment, the surveillance system operates without recording data.

Once the model is fitted, critical social density is identified as:

\[
\rho_c = \rho_{\text{pred}}^{\text{low}},
\]

(7)

where \( \rho_{\text{pred}}^{\text{low}} \) is the lower bound of the 95% prediction interval \((\rho_{\text{pred}}^{\text{low}}, \rho_{\text{pred}}^{\text{ub}})\) at \( v = 0 \), as illustrated in Figure 2.

Keeping \( \rho \) under \( \rho_c \) will keep the probability of social distancing violation occurrence near zero with the linear regression assumption.

5. Experiments

We conducted 3 case studies to evaluate the proposed method. Each case utilizes a different pedestrian crowd dataset. They are Oxford Town Center Dataset (an urban street) \(^{[39]}\), Mall Dataset (an indoor mall) \(^{[40]}\), and Train Station Dataset (New York City Grand Central Terminal) \(^{[41]}\). Table \[I\] shows detailed information about these datasets.

5.1. Implementation details

The first step was finding the perspective transformation matrix \( \mathbf{M} \) for each dataset. For Oxford Town Center Dataset, we directly used the transformation matrix available on its official website. For Train Station Dataset, we found the floor plan of NYC Grand Central Terminal and measured the exact distances among some key points that were used for calculating the perspective transformation. For Mall Dataset, we first estimated the size of a reference object in the image by comparing it with the width of detected pedestrians and then utilized the key points of the reference object to calculate the perspective transformation.

The second step was applying the pedestrian detector on each dataset. The experiments were conducted on a regular PC with an Intel Core i7-4790 CPU, 32GB RAM, and an Nvidia GeForce GTX 1070Ti GPU running Ubuntu 16.04 LTS 64-bit operating system. Once the pedestrians were
detected, their positions were converted from the image coordinates into the real-world coordinates.

The last step was conducting social distancing monitoring and finding the critical density $\rho_c$. Only the pedestrians within the ROI were considered. The statistics of the social density $\rho$, the inter-pedestrian distances $d_{i,j}$, and the number of violations $v$ were recorded over time. The analysis of statistics is described in the following section.

6. Results

6.1. Pedestrian Detection

We experimented with two different deep CNN based object detectors: Faster R-CNN and YOLOv4. Figure 3 shows the pedestrian detection results using Faster R-CNN [9] and the corresponding social distancing in world coordinates. The detector performances are given in Table 2. As can be seen in the table, both detectors achieved real-time performance. In terms of detection accuracy, we provide the results of MS COCO dataset from the original works [9][10].

6.2. Social Distancing Monitoring

For pedestrian $i$, the closest physical distance is $d_{i, j}^{\text{min}} = \min(d_{i,j}), \forall j \neq i \in \{1, 2, \cdots n\}$. Based on $d_{i, j}^{\text{min}}$, we further calculated two metrics for social distancing monitoring: the minimum closest physical distance $d_{\text{min}}^{\text{social}} = \min(d_{i, j}^{\text{min}}), \forall i, j \neq i \in \{1, 2, \cdots n\}$. The analysis of these metrics is described in the following section.
min\(d_{\text{min}}\), \(\forall i \in \{1, 2, \cdots, n\}\) and the average closest physical distance \(d_{\text{avg}} = \frac{1}{n} \sum_{i=1}^{n} d_{\text{min}}\). Figure 7 shows the change of \(d_{\text{min}}\) and \(d_{\text{avg}}\) as time evolves. They are compared with the social density \(\rho\). From the figure we can see that when \(\rho\) is relatively low, both the \(d_{\text{min}}\) and \(d_{\text{avg}}\) are relatively high, for example, \(t = 85s\) in Train Station Dataset, \(t = 80s\) in Mall Dataset, and \(t = 100s\) in Oxford Town Center Dataset. This shows a clear negative correlation between \(\rho\) and \(d_{\text{avg}}\). The negative correlation is further visualized as 2D histograms in Figure 5.

### 6.3. Critical Social Density

To find the critical density \(\rho_c\), we first investigated the relationship between the number of social distancing violations \(v\) and the social density \(\rho\) in 2D histograms, as shown in Figure 6. As can be seen in the figure, \(v\) increases with an increase in \(\rho\), which indicates a linear relationship with a positive correlation.

Then, we conducted the simple linear regression, using the regression model of equation (6), on the data points of \(v\) versus \(\rho\). The skewness values of \(\rho\) for Oxford Town Center Dataset, Mall Dataset, and Train Station Dataset are 0.25, 0.16, and -0.07, respectively, indicating the distributions of \(\rho\) are symmetric. This satisfies the normality assumption of the error term in linear regression. The regression result is displayed in figure 4. The critical density \(\rho_c\) was identified as the lower bound of the prediction interval at \(v = 0\).

Table 3 summaries the identified critical densities \(\rho_c\) as well as the intercepts \(\beta_0\) of the regression models. The obtained critical density values for all datasets are similar. They also follow the patterns of the data points as illustrated in Figure 4. This verified the effectiveness of our method.

### 7. Conclusion

This work proposed an AI and monocular camera based real-time system to monitor the social distancing. We use the proposed critical social density to avoid overcrowding by modulating inflow to the ROI. The proposed method was verified using 3 different pedestrian crowd datasets.

There are some missing detections in the Train Station Dataset, as in some areas the pedestrian density is extremely high and occlusion happens. However, after some qualitative analysis, we concluded that most of the pedestrians were captured and the idea of finding critical social density is still valid.

Pedestrians who belong to a group were not considered as a group in the current work, which can be a future direction. Nevertheless, one may argue that even individuals
who have close relationships should still try to practice social distancing in public areas.

References

[1] C. Courtemanche, J. Garuccio, A. Le, J. Pinkston, and A. Yelowitz, “Strong social distancing measures in the united states reduced the covid-19 growth rate: Study evaluates the impact of social distancing measures on the growth rate of confirmed covid-19 cases across the united states.” Health Affairs, pp. 10–1377, 2020.

[2] C. T. Nguyen, Y. M. Saputra, N. Van Huynh, N.-T. Nguyen, T. V. Khoa, B. M. Tuan, D. N. Nguyen, D. T. Hoang, T. X. Vu, E. Dutkiewicz et al., “Enabling and emerging technologies for social distancing: A comprehensive survey,” arXiv preprint arXiv:2005.02816, 2020.

[3] S. Agarwal, N. S. Punn, S. K. Sonbhadra, P. Nagabushan, K. Pandian, and P. Saxena, “Unleashing the power of disruptive and emerging technologies amid covid 2019: A detailed review,” arXiv preprint arXiv:2005.11507, 2020.

[4] N. S. Punn, S. K. Sonbhadra, and S. Agarwal, “Monitoring covid-19 social distancing with person detection and tracking via fine-tuned yolo v3 and deepsort techniques,” arXiv preprint arXiv:2005.01385, 2020.

[5] M. Cristani, A. Del Bue, V. Murino, F. Setti, and A. Vinciarrelli, “The visual social distancing problem,” arXiv preprint arXiv:2005.04813, 2020.

[6] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, “Privacy preserving crowd monitoring: Counting people without people models or tracking,” in 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2008, pp. 1–7.

[7] F. Z. Qureshi, “Object-video streams for preserving privacy in video surveillance,” in 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance. IEEE, 2009, pp. 442–447.

[8] S. Park and M. M. Trivedi, “A track-based human movement analysis and privacy protection system adaptive to environmental contexts,” in IEEE Conference on Advanced Video and Signal Based Surveillance, 2005. IEEE, 2005, pp. 171–176.

[9] 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.

[10] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” arXiv preprint arXiv:2004.10934, 2020.

[11] D. S. Hui, E. I. Azhar, T. A. Madani, F. Ntoumi, R. Kock, O. Dar, G. Ippolito, T. D. Mchugh, Z. A. Memish, C. Drosten et al., “The continuing 2019-ncov epidemic threat of novel coronaviruses to global healththe latest 2019 novel coronavirus outbreak in wuhan, china,” International Journal of Infectious Diseases, vol. 91, pp. 264–266, 2020.
Figure 7. The change of minimum closest physical distance $d_{\text{min}}$, average closest physical distance $d_{\text{avg}}$, and the social density $\rho$ over time.

[12] M. Greenstone and V. Nigam, “Does social distancing matter?” University of Chicago, Becker Friedman Institute for Economics Working Paper, no. 2020-26, 2020.

[13] Z. Zou, Z. Shi, Y. Guo, and J. Ye, “Object detection in 20 years: A survey,” arXiv preprint arXiv:1905.05055, 2019.

[14] 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.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[16] R. Girshick, “Fast r-cnn,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.

[17] J. Dai, Y. Li, K. He, and J. Sun, “R-fcn: Object detection via region-based fully convolutional networks,” in Advances in neural information processing systems, 2016, pp. 379–387.

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

[19] J. Redmon and A. Farhadi, “Yolo9000: better, faster, stronger,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7263–7271.
[20] ——, “Yolov3: An incremental improvement,” arXiv preprint arXiv:1804.02767, 2018.

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in European conference on computer vision. Springer, 2016, pp. 21–37.

[22] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

[23] M. Tan, R. Pang, and Q. V. Le, “Efficientdet: Scalable and efficient object detection,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10781–10790.

[24] H. Law and J. Deng, “Cornernet: Detecting objects as paired keypoints,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 734–750.

[25] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, “Centernet: Keypoint triplets for object detection,” in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 6569–6578.

[26] Z. Tian, C. Shen, H. Chen, and T. He, “Fcos: Fully convolutional one-stage object detection,” in Proceedings of the IEEE international conference on computer vision, 2019, pp. 9627–9636.

[27] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, “Reppoints: Point set representation for object detection,” in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 9657–9666.

[28] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” International journal of computer vision, vol. 88, no. 2, pp. 303–338, 2010.

[29] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge: A retrospective,” International journal of computer vision, vol. 111, no. 1, pp. 98–136, 2015.

[30] 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. Springer, 2014, pp. 740–755.

[31] LandingAI, Landing AI Creates an AI Tool to Help Customers Monitor Social Distancing in the Workplace, 2020 (accessed May 3, 2020). [Online]. Available: https://landing.ai/landing-ai-creates-an-ai-tool-to-help-customers-monitor-social-distancing-in-the-workplace/

[32] P. Khandelwal, A. Khandelwal, and S. Agarwal, “Using computer vision to enhance safety of workforce in manufacturing in a post covid world,” arXiv preprint arXiv:2005.05287, 2020.

[33] J. Hall, Social Distance Monitoring, 2020 (accessed June 1, 2020). [Online]. Available: https://levefive.com/social-distance-monitoring/