Video-based Social Distancing: Evaluation in the COSMOS Testbed

Mahshid Ghasemi, Zhengye Yang, Mingfei Sun, Hongzhe Ye, Zihao Xiong, Javad Ghaderi, Zoran Kostic, and Gil Zussman

***Abstract*—Social distancing is an effective public health tool to reduce the spread of respiratory pandemics such as COVID-19. To analyze compliance with social distancing policies, we design two video-based pipelines for social distancing analysis, namely, *Auto-SDA* and *B-SDA*. Auto-SDA (Automated video-based Social Distancing Analyzer) is designed to measure social distancing using street-level cameras. To avoid privacy concerns of using street-level cameras, we further develop B-SDA (Bird's eye view Social Distancing Analyzer), which uses bird's eye view cameras, thereby preserving pedestrian's privacy. We used the COSMOS testbed deployed in West Harlem, New York City, to evaluate both pipelines. In particular, Auto-SDA and B-SDA are applied on videos recorded by two of COSMOS cameras deployed on the 2nd floor (street-level) and 12th floor (bird's eye view) of Columbia University's Mudd building, looking at 120th St. and Amsterdam Ave. intersection, New York City. Videos are recorded before and during the peak of the pandemic, as well as after the vaccines became broadly available. The results represent the impact of social distancing policies on pedestrians' social behavior. For example, the analysis shows that after the lockdown, less than 55% of the pedestrians failed to adhere to the social distancing policies, whereas this percentage increased to 65% after the vaccines' availability. Moreover, after the lockdown, 0-20% of the pedestrians were affiliated with a social group, compared to 10-45% once the vaccines became available. The results also show that the percentage of face-to-face failures has decreased from 42.3% (pre-pandemic) to 20.7%(after the lockdown).**

***Index Terms*—Social distancing, COVID-19, object detection, tracking, smart city, testbeds.**

I. Introduction

Social distancing has been proven to be an effective tool in reducing the spread of respiratory pandemics such as COVID-19. Traditionally, compliance with preventive measures such as social distancing policies has been evaluated through survey-based methods [4], [5]. Such approaches necessitate the engagement and cooperation of individuals, making the process potentially inefficient and time-consuming. Alternatively, manual observation of video footage has been utilized to assess individuals' adherence to these policies [6]. However, these methods may introduce bias and inaccuracies due to observers' subjectivity and fatigue. With the growing prevalence of traffic cameras and advances in artificial intelligence (AI) technologies, the opportunity for automation emerges, promising both enhanced efficiency and accuracy. By leveraging these technologies, it is possible to automate compliance measurement, reducing human bias and error and to offer a more timely analysis.

Therefore, in this work, we present two pipelines for social distancing analysis based on video cameras, namely, **Auto**mated video-based **S**ocial **D**istancing **A**nalyzer (**Auto-SDA**) and **B**ird's eye view **S**ocial **D**istancing **A**nalyzer (**B-SDA**), which are designed to measure pedestrians' compliance with social distancing policies using street-level and bird's eye view cameras, respectively. Auto-SDA offers high accuracy, which is not sensitive to the dynamics of the scene and the camera's tilt angle. On the other hand, B-SDA provides comparable accuracy while preserving pedestrians' privacy.

Challenges. Our video-based social-distancing pipelines use off-the-shelf models for pedestrian detection and tracking. However, achieving highly accurate social-distancing analysis requires overcoming several challenges, as outlined below:

- **Distance measurement:** To accurately measure the distance between pedestrians, it is necessary to convert 2D pixel distances into real-world 3D distances on the ground (this process is called calibration). Calibration is more challenging for street-level cameras due to their oblique view compared to bird's eye cameras that have a top-level view.
- **Pedestrian tracking:** Due to the moving vehicles and static obstacles on the road, such as traffic lights, the tracker model might miss pedestrians or assign multiple IDs to a single pedestrian. Rectifying the output of tracker models is particularly important for street-level cameras since their lower viewpoint increases the likelihood of occlusion. Conversely, for bird's eye cameras, with their high view that minimizes occlusion, rectifying tracker outputs is less crucial.
- **Group detection:** Identifying and distinguishing affiliated pedestrians (i.e., pedestrians who walk together as a social group) is very important. These groups need to be recognized and excluded from social distancing failures.
- **Camera perspective:** Handling different camera perspectives, such as bird's eye view, brings its own set of challenges. For instance, from such perspectives, the smaller appearance of pedestrians can make detection more difficult. These complexities can be mitigated during the preprocessing phase.

Contributions. To cope with the above challenges and achieve

This work was supported in part by ARO grant W911NF1910379, NSF grants CNS-1827923 and OAC-2029295, NSF grant CNS-2038984 and corresponding support from the Federal Highway Administration (FHA), and NSF grant CNS-2148128 and by funds from federal agency and industry partners as specified in the Resilient & Intelligent NextG Systems (RINGS) program. Partial and preliminary results appeared in [1]–[3].

Mahshid Ghasemi, Mingfei Sun, Hongzhe Ye, Zihao Xiong, Javad Ghaderi, Zoran Kostic, and Gil Zussman are with the Department of Electrical Engineering, Columbia University, New York, NY 10027, USA (e-mail: {mahshid.ghasemi, ms5898, hy2610, zx2273, jghaderi, zk2172, gil.zussman}@columbia.edu).

Zhengye Yang is with the Department of Computer, and Systems Engineering, RPI, Troy, NY 12180, USA (e-mail: yangz15@rpi.edu).

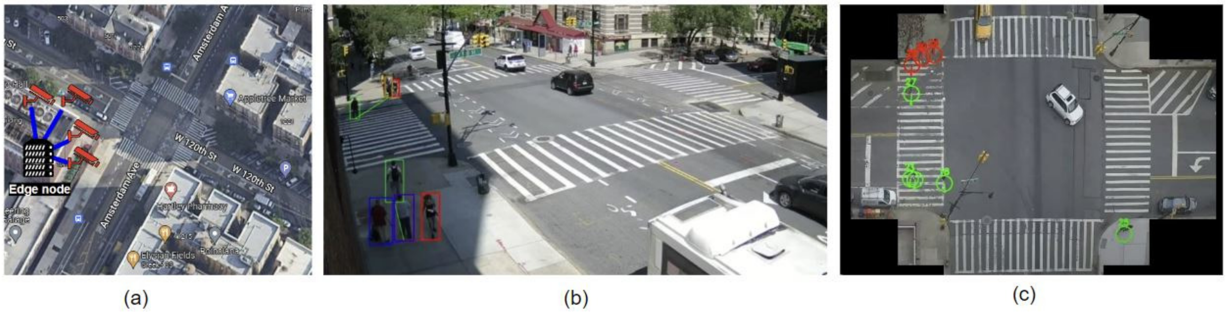

Fig. 1: (a) The NSF PAWR COSMOS site at 120th St. and Amsterdam Ave. intersection, NYC; Social distancing evaluation using (b) Auto-SDA and (c) B-SDA (Green: distance $>$ 6 ft; Blue: social group; Red: distance $<$ 6 ft).

the necessary accuracy, both systems require the addition of specifically designed modules to the off-the-shelf pedestrian detection and tracking models. Accordingly, the key contributions of our work are summarized below:

- We designed and incorporated three modules into Auto-SDA: (i) a *calibration module* that converts 2D pixel distances into 3D on-ground distances with *less than 10 cm* error, (ii) a *correction module* that identifies pedestrians who were missed or assigned duplicate IDs by the tracker and rectifies their trajectories, and (iii) a *group detection module* that identifies affiliated pedestrians and excludes them from social distancing failure analysis.
- In the B-SDA system, we incorporated pre-processing modules that enable social distancing analysis from a bird's eye view camera perspective. B-SDA also includes the same group detection module as Auto-SDA. Similarly to Auto-SDA, B-SDA is also equipped with a *group detection module*.
- We fine-tuned and evaluated the two pipelines using real-world data recorded by two cameras as part of the COSMOS testbed deployed in West Harlem, New York City (NYC) [7]–[9]. The cameras are installed on the 2nd floor (street-level) and 12th floor (bird's eye view) of Columbia's Mudd building, looking at 120th St. and Amsterdam Ave. intersection, NYC (see Fig. 1(a)). Fig. 1(b) demonstrates social distancing analysis using Auto-SDA, and Fig. 1(c) demonstrates B-SDA's evaluation of social distancing failures.[1]
- To assess the impact of social distancing policies on pedestrians' social behavior, we applied Auto-SDA and B-SDA on videos recorded by these cameras, before the COVID-19 outbreak, soon after the lockdown and after the vaccines became broadly available.
  i) Auto-SDA's dataset consists of 180 sec videos recorded at different times of the day (9 AM, 2 PM, 5:30 PM, 7:30 PM, and 10 PM) in about one month periods, soon after the lockdown (June 17 to July 20, 2020), and after the vaccines became broadly available (May 2021). In addition, the dataset includes 16 videos collected (less methodically) before the pandemic (June 2019), which are used as a reference point.
  ii) B-SDA's dataset consists of 300 sec videos recorded multiple times per day (9 AM, 2 PM, 5:30 PM, and 10 PM) from June 2020 to February 2021. It also includes sporadically collected videos between June and July 2019 (prior to the pandemic).

The results of applying Auto-SDA/B-SDA on recorded videos show that after the lockdown, less than 55% of the pedestrians failed to comply with the social distancing protocols compared to 65% post-vaccine. The results also show that the fraction of pedestrians walking as a social group has grown from 0-20% (after the lockdown) to 10-45% (post-vaccine). We also compared the duration of social distancing failures during the pandemic and post-vaccine periods, with results indicating a statistical increase in the duration of social distancing failures in the post-vaccine period. Furthermore, the results suggest a significant decrease in pedestrian density after the lockdown (compared to pre-pandemic), while the density has slightly increased after the availability of vaccines. Moreover, the percentage of face-to-face failures has decreased from 42.3% (pre-pandemic) to 20.7%(after the lockdown).

The observed trend in social distancing compliance, from before the pandemic via during the pandemic to the post-vaccine availability period, is consistent with the findings reported in various U.S. surveys [5], [10], [11] conducted during the same time frame. To the best of our knowledge, this is the first work to provide statistics regarding the evolution of social distancing compliance using automated video-based social distancing analyzers.

Organization. The rest of the paper is organized as follows. In Sec. II, we present the research background and related work on using computer vision to monitor social distancing. Sections III and IV describe the implementation of Auto-SDA and B-SDA pipelines. In Sec. V, we present the results of applying Auto-SDA and B-SDA on our dataset. Conclusion and future plans are discussed in Sec. VI and VII, respectively.

II. Background and Related Work

Object detection is a computer-vision technique for locating instances of objects in images or videos. Most state-of-the-art object detectors are deep learning-based. Among the prominent approaches, R-CNN [12], Fast R-CNN [13], Faster R-CNN [14], and Mask R-CNN [15] use a two-stage structure for object detection, which consists of region proposal stage and classification stage. In contrast, SSD [16] and YOLO methods [17]–[20] have a single-stage structure, achieving higher inference speeds. Improvements of YOLO [18]–[20] lead to a detection accuracy comparable to R-CNN without sacrificing YOLO's inference speed in our scenario. Consider-

[1] Sample videos of Auto-SDA and B-SDA evaluation appear in https://bit.ly/3ZqznGb and https://bit.ly/3yGD0Mw, respectively.

ing the speed-accuracy trade-off, both Auto-SDA and B-SDA use YOLOv4 [20] object detection model to detect pedestrians.

In addition to an object detector, a tracker is required to extract the trajectory of each pedestrian and measure the duration in which it has been in close proximity to an unaffiliated pedestrian. Auto-SDA uses the NVIDIA DCF tracker, which leverages a Discriminative Correlation Filter (DCF) based approach for visual object tracking and the Hungarian algorithm for data association. Based on our experiments, DCF provides higher accuracy than other trackers, such as DeepSORT [21] (an extension to the Simple Online and Realtime Tracking SORT algorithm) and DAN (Deep Affinity Network) that jointly learns target object appearances and their affinities in a pair of video frames in an end-to-end fashion [22].

Table I summarizes the main features of Auto-SDA and B-SDA compared to the prior work focusing on monitoring the COVID-19 pandemic. Surveys on technologies for monitoring social distancing and pandemic management appear in [23]–[25]. The social distancing framework proposed in [26] uses MobileNet Single Shot Multibox Detector (SSD) model for human detection. It then compares the pixel distances between individuals with a fixed value to distinguish social distancing failures. However, if the camera is not perpendicular to the ground, assuming a fixed threshold on pixel distances leads to inaccurate failure detection.

The study [27] proposes the use of monocular cameras and deep learning-based object detectors to monitor social distancing and emit warnings. However, since it does not use a tracker, it can only provide instantaneous warnings. In [28], a deep learning detection technique based on YOLOv2 is proposed. It uses thermal images to detect people and verify their compliance with social distancing. The platforms [27], [28] use homography transformation to convert 2D on-image coordinates to their 3D counterparts, which can only be used to estimate the camera pose for planar objects and are not accurate enough for street-level views. Thus, a more advanced method is required to calibrate the cameras and compute the on-ground distances from the pixel distances on an image. The framework in [29] uses YOLOv3 for object detection and DeepSORT for tracking. The obtained bounding boxes are utilized to obtain depth information of the pedestrians (i.e., their distance from the camera lens) and identify clusters of pedestrians neglecting social distancing. However, the depth information-based method is not sufficiently accurate for measuring the distance between pedestrians, and a more precise camera calibration along with group detection is needed.

The platform proposed in [30] performs human head detection on UAV (unmanned aerial vehicles) images to locate pedestrians. It then calculates the distance between detected pedestrians to verify social distancing compliance. The frameworks in [30]–[35] employ an object detector but do not use a tracker to derive trajectories. Moreover, they perform planar camera perspective transformation for calibration, which yields an inaccurate estimation of the on-ground coordinates, thereby limiting the social distancing measurements' accuracy.

The platforms discussed above only use street-level cameras and do not provide evaluations on real-world data recorded during the COVID-19 pandemic. In this work, we study the usage of bird's-eye view cameras as well as street-level cameras to measure social distancing compliance. Moreover, we applied Auto-SDA and B-SDA on videos recorded during the COVID-19 pandemic and measured the impacts of the outbreak on pedestrians' social behavior.

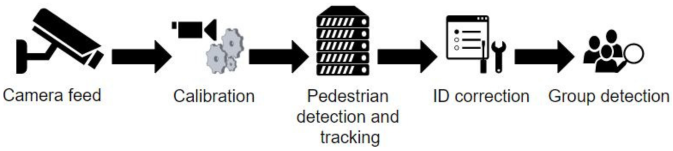

Fig. 2: Different stages in the Auto-SDA pipeline.

III. Auto-SDA Pipeline

Auto-SDA is designed to be a highly accurate social distancing analyzer pipeline whose performance is not sensitive to the camera's tilt-angle and scene dynamics. The pipeline consists of multiple modules (see Fig. 2), including an object detection module (YOLOv4 [36]) and a tracking module (Nvidia DCF-based tracker). While these are off-the-shelf components, achieving high accuracy calls for the design of tailored components. Specifically, we incorporated three modules in Auto-SDA, as outlined below.

- **Camera calibration module:** Our measurements show that using a single set of photogrammetry parameters for the whole scene leads to imprecise on-ground distance computation. Therefore, this module breaks the view of the camera into multiple areas and computes the corresponding photogrammetry parameters for each area individually. These parameters are then used to convert the 2D on-image distances into 3D on-ground distances with less than 10 cm error.
- **ID correction module:** This module compensates for the inaccuracies of the object detector and tracking model caused by the camera's tilt angle and the obstacles on the road. For instance, if multiple IDs are assigned to a single pedestrian, this module removes the redundant IDs and derives the entire trajectory of that pedestrian.
- **Group detection module:** This module detects the pedestrians affiliated with a single social group (e.g., members of a family) and excludes them from social distancing failure.

In the following, we describe each module in detail.

A. Camera Calibration

Camera calibration is a necessary step for extracting on-ground distances between pedestrians. The goal is to determine the intrinsic and extrinsic parameters of the camera to convert the 2D on-image coordinates viewed by the camera to the 3D on-ground coordinates. Intrinsic parameters are (i) principal point (c_x, c_y), (ii) focal length in pixel units (f_x, f_y), (iii) radial distortion coefficients $(k_1, k_2, ..., k_6)$, and (iv) tangential distortion coefficients (p_1, p_2). Extrinsic parameters are (i) rotation matrix R, and (ii) translation vector t.[2]

Since the COSMOS cameras are fixed, we needed to calibrate them only once. To do so, we captured multiple

[2] The effects of higher order coefficients are negligible, see [37].

TABLE I: A comparison of prior work on social distancing analysis using street-level cameras to Auto-SDA

Framework	Object Detection	Tracking	Calibration Method	On-Ground Distance Computation Error	Correction	Group Detection	Real-World COVID-19 Pandemic Impact Analysis
[27]	✓	X	Homography trans.	$\gg$ 10 cm	X	X	X
[29]	✓	✓	Depth information	$\gg$ 10 cm	X	X	X
[30]–[35]	✓	X	Planar camera persp. trans.	$\gg$ 10 cm	X	X	X
[26]	✓	X	Fixed scaling	$\gg$ 10 cm	X	X	X
Auto-SDA	✓	✓	Multi-area calibration	$<$ 10 cm	✓	✓	✓
B-SDA	✓	✓	Planar camera persp. trans.	$<$ 10 cm	N/A due to bird's eye view	✓	✓

TABLE II: A comparison of calibration methods used in the prior work to Auto-SDA's multi-area calibration

Pixel Coordinates of a Pair of Points on a 4 K Frame	On-Ground Distance (cm)	Distance Calculated by Multi-area Calibration (cm)	Distance Calculated by Homography Trans. [27] (cm)	Distance Calculated by Planar Camera Persp. Trans. [30]–[35] (cm)
[1093, 715], [1065, 685]	320	325	209	339
[1785, 572], [1862, 566]	183	178	140	128
[1680, 582], [1588, 552]	503	508	368	457
[2153, 598], [2077, 582]	259	256	201	146
[1121, 746], [1093, 714]	320	314	201	229

photos of a checkerboard with known square sizes, posed in different tilt and rotation angles (see Fig. 3). To calculate the intrinsic parameters, we fed the 2D on-image pixel coordinates of the checkerboard corners and their corresponding 3D coordinates into OpenCV [38], which runs the global Levenberg-Marquardt optimization algorithm [39].

We split the view of the intersection into 10 areas (as shown in Fig. 4) and we determined the extrinsic parameters for each area. This can further mitigate the impact of camera distortion and obtain the on-ground distances with less than 10 cm error (ground truth is obtained from actual distance measurements in the intersection). Accuracy improves with the number of areas, but for our use case, 10 areas proved to be adequate. For each area, we selected a few points on the ground with known 3D coordinates and found their corresponding 2D pixel coordinates in the camera's view. These sample points, along with the intrinsic parameters of the camera, are then used to determine the extrinsic parameters (using OpenCV).

Auto-SDA plugs these parameters into the photogrammetry equations [38], [40], [41], given below, and completes the 2D-3D coordinates conversion:

$$\begin{aligned}
&\begin{bmatrix} x & y & z \end{bmatrix}^T = \boldsymbol{R}\begin{bmatrix} X & Y & Z \end{bmatrix}^T + t, \quad x' = \frac{x}{z}, \quad y' = \frac{y}{z} \\
&x'' = x' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 x' y' + p_2(r^2 + 2x'^2) \\
&y'' = y' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + p_1(r^2 + 2y'^2) + 2p_2 x' y' \\
&r^2 = x'^2 + y'^2, \quad u = f_x x'' + c_x, \quad v = f_y y'' + c_y.
\end{aligned}$$

In the equations above, $[u, v]$ are the 2D pixel coordinates, and $[X, Y, Z]$ are the 3D on-ground coordinates. Since there are no closed-form equations to map the 2D points to 3D points, Auto-SDA uses Newton's method to solve the above system of equations (it sets the ground level to $Z = 0$ and solves for X and Y).

In Table II, we compare the accuracy of on-ground distance calculation of the multi-area calibration method used in Auto-SDA with the calibration methods used in [27], [31]–[34]. As the results show, there could be more than 1 m error in calculating the on-ground distances when using the homography and planar camera perspective transformation method used in the prior work. While such accuracy may be sufficient for other applications, it is clearly inadequate for social distancing monitoring. Moreover, in [29], the distance between pedestrians is determined by using a method proposed in [42], which results in poor accuracy. In this method, first, the distance of a pedestrian from the camera lens is obtained using the coordinates, width, and height of its bounding box provided by an object detector. Then, the distance between every two pedestrians is calculated. In Fig. 5, we represent the results of calculating the pedestrians' distances from the camera lens using the calibration method proposed in [42]. The camera's (vertical and horizontal) distance from the pedestrians is more than 10 m. However, due to the oblique view of the camera, the calculated distances (displayed near the bounding boxes) are far from their true values, and one cannot simply fix them (e.g., using a scaling factor).

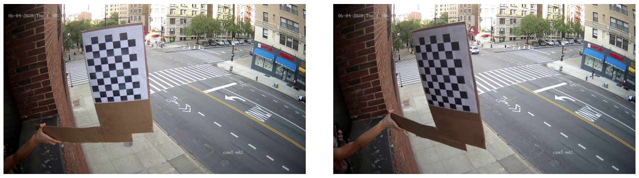

Fig. 3: Calibration of the COSMOS cameras using a checkerboard: more than 20 images of the checkerboard in different poses were provided to the OpenCV library to obtain the intrinsic parameters of the camera.

B. Pedestrian Detection and Tracking

Auto-SDA uses the YOLOv4 object detector [36] to detect pedestrians. It is also equipped with a tracker (NvDCF) that extracts the trajectory of each pedestrian and uses that to trace the number of pedestrians with whom he/she is in contact (within a radius of 6 ft) and the duration of each contact. Both models are set as building blocks inside the Deepstream

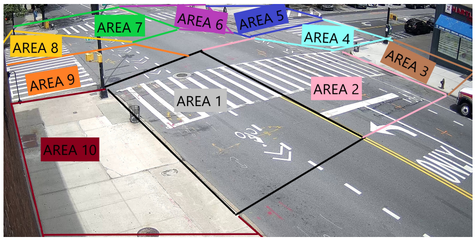

Fig. 4: Division of the camera scene into 10 areas. The extrinsic parameters of the camera were calculated for each area individually.

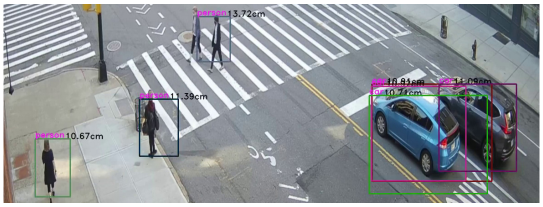

Fig. 5: Computed distances of pedestrians from the camera using objects' depth information proposed in [42]. The real distances (both vertically and horizontally) of the pedestrians from the camera are each greater than 10 m. However, due to the oblique view of the camera, the obtained distances deviate from their real values and it is not straightforward to rectify them (e.g., using a scaling factor.)

pipeline which is an optimized architecture built using the Gstreamer framework [43].

C. ID Correction

The COSMOS street-level camera is located at a corner of the intersection and has an oblique view of the area. Therefore, the pedestrians are small and might be blocked by obstacles such as vehicles, traffic lights, and other pedestrians. As a result, the tracker may lose a pedestrian along the way or assign multiple IDs to a single person, leading to degraded performance.

The ID Correction module is designed to mitigate this. It detects the IDs that belong to a single pedestrian and extracts their entire trajectory. Algorithm 1 describes our ID Correction algorithm. It receives the results of the object detector and tracking module as its input, and, for each ID, it creates a structure in which it keeps the trajectory ($id.Trj$), the first and last time it was detected by the tracker ($id.TimeStamp$), and the parameters of the Linear Regression approximation of the tail and head of the trajectory. The algorithm then uses this information to predict the pedestrian's location before it was detected and after it was lost.

For each ID pair (id_1, id_2), the ID Correction algorithm then verifies three conditions to determine whether they are associated with a single pedestrian or not. First, the gap between id_1 lost time, t_1, and id_2 detected time, t_2, must be small enough (less than a predefined threshold e_1). Second, the distance between the predicted location of id_1 at time t_2 (based on the Linear Regression approximation for the tail of id_1 trajectory) and the location of id_2 at time t_2 has to be less than a specified threshold (e_2). Third, it measures the angle between id_1's tail direction and id_2's head direction. This angle must be less than 90° to ensure that the algorithm does not mistake two pedestrians crossing each other in opposite directions for a single pedestrian. If all three conditions hold, then id_1 and id_2 belong to a single person. An example is shown in Fig. 6, where the tracker has assigned three IDs to a single pedestrian. The ID correction module detects the segments that belong to a single trajectory by using the Linear Regression approximation corresponding to the tail of each segment and comparing the estimated start point and the real start point of the subsequent segment.

Algorithm 1 ID Correction

```
Input: $ID_{\text{vec}}, e_1, e_2, n$                    ▷ $ID_{\text{vec}}$ is the output of NvDCF tracker
Output: corrected $ID_{\text{vec}}$
for $id \in ID_{\text{vec}}$ do
    Compute $id.Trj$                                     ▷ vector of points on id's path
    Compute $id.TimeStamp.StartTime$                     ▷ detection time
    Compute $id.TimeStamp.StopTime$                      ▷ lost time
    Compute $(id.TailEst, id.TailDir)$                   ▷ Linear Regression of $id.Trj.tail(n)$
    Compute $(id.HeadEst, id.HeadDir)$                   ▷ Linear Regression of $id.Trj.head(n)$
end for
for $(id_1, id_2) \in ID_{\text{vec}}$ do
    $t1 \leftarrow id_1.TimeStamp.StopTime$
    $t2 \leftarrow id_2.TimeStamp.StartTime$
    $p_1 \leftarrow id_1.TailEst.\text{at}(t = t_2), \quad p_2 \leftarrow id_1.Trj.\text{at}(t_2)$
    $v_1 \leftarrow id_1.TailDir, \quad v_2 \leftarrow id_2.HeadDir$
    if $t_2 - t_1 < e_1$ && $|p_1 - p_2| < e_2$ && $\angle(v1, v2) < 90$ then
        $id_1$ and $id_2$ belongs to same person
    end if
end for
```

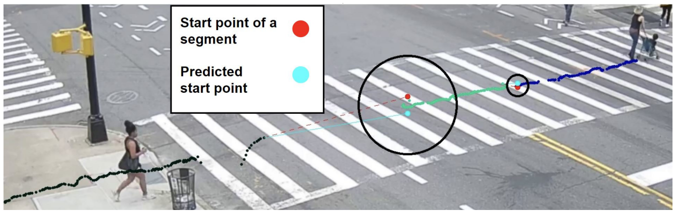

Fig. 6: Demonstration of detection and removal of redundant IDs by the ID Correction algorithm when the tracker assigns 3 IDs to a single pedestrian.

D. Group Detection

We enhance the social distancing analysis by distinguishing the pedestrians walking together as a social group (e.g., friends/family) and excluding them from social distancing failures. There are several methods proposed for group detection, e.g., see [44]–[47]. All these group detection methods require details such as velocity, body and head orientation, and exact trajectory. However, in our setting (and in many realistic deployments), the cameras are mounted at a relatively high altitude, viewing the intersection from a corner with a large tilt angle. Moreover, various obstacles on the road might block the view of pedestrians for some periods. Therefore, such detailed information cannot be obtained from these cameras.

We designed a group detection algorithm that can detect pedestrians belonging to a single social group with the limited data we can derive from cameras such as the ones in the COSMOS testbed. The Group Detection algorithm is given in Algorithm 2. It uses the IDs of the pedestrians rectified in the ID Correction module to derive an approximation of each pedestrian trajectory. Then, it calculates the correlation between these trajectories to check if two pedestrians belong to a single social group. Specifically, the algorithm calculates the distance between each pair of pedestrians (id_1, id_2) on all the frames and then calculates the average distance $(\bar{d})$ and

Algorithm 2 Group Detection

```
Input: $ID_{\text{vec}}, d_{\text{max}}, d_{\text{max}}, \sigma_{\text{max}}$
Output: $ID_{\text{vec}}$ Pedestrians belong to a group
for $id \in ID_{\text{vec}}$ do
    $id.TimeTrj = \text{map}(id.TimeStepVec, id.Trj)$
end for
for $(id_1, id_2) \in ID_{\text{vec}}$ do
    $n = 0$
    for $t = 1 : T$ do
        $pos_1 = id_1.TimeTrj(t), \quad pos_2 = id_2.TimeTrj(t)$
        $d = ||pos_1 - pos_2||_2$
        if $d > d_{\text{max}}$ then
            $n++$, continue
        end if
        $Corr_{\text{vec}}(id_1, id_2).\text{append}(d)$
    end for
    if $n > N_{\text{max}}$ then
        continue
    end if
    $\bar{d} = \text{mean}(Corr_{\text{vec}}(id_1, id_2))$  ▷ calculate the mean distance between two pedestrians
    $\sigma = \text{std}(Corr_{\text{vec}}(id_1, id_2))$  ▷ calculate the standard deviation of instantaneous distances between two pedestrians
    if $\bar{d} < \Delta_{\text{max}}$ && $\sigma < \sigma_{\text{max}}$ then
        $id_1$ and $id_2$ belongs to the same group
    end if
end for
```

empirical standard deviation (σ). In line with previous research on investigating various metrics for group identification by studying the social nature of human behavior [48], [49], a pair of pedestrians is labeled as one social group under two main conditions:

- Their instantaneous distance (d) does not exceed the maximum distance (d_{max}) in more than the maximum allowed frames (N_{max}).
- The mean and standard deviation of their distance fall below the maximum average distance (Δ_{max}) and the maximum standard deviation (σ_{max}), respectively.

The algorithm's parameters (i.e., d_{max}, N_{max}, Δ_{max}, and σ_{max}) were fine-tuned using recorded sample videos from the COSMOS street-level camera. It was then evaluated on three 10-minute sample videos from the same camera, achieving over 85% accuracy in group detection compared to visually detected social groups.

IV. B-SDA Pipeline

Although Auto-SDA can detect social distancing failures with high accuracy, using street-level cameras has several challenges. For example, the surveillance area of street-level cameras is limited, tracking of pedestrians is challenging due to occlusions, and face/license plate recognition can raise privacy concerns. In B-SDA, we have addressed these challenges by using bird's eye view cameras. To facilitate successful measurement of social distancing using bird's-eye camera recordings, two preliminary steps are required: (i) per-frame detection of pedestrians within the scene [12]–[20], and (ii) reliable tracking of pedestrian trajectories across video frames [21], [50]–[52]. The size of pedestrians in bird's-eye view videos is a function of video resolution and can be smaller than 30×30 pixels for 1080p recordings. Processing such small objects is a challenge for conventional object detection and tracking algorithms. To mitigate these issues, as illustrated in Fig. 7, B-SDA incorporates the following modules.

- **Data pre-processing module:** Contains three components: Weighted-Mask Background Subtraction, Video Calibration, and Center Cropping. This module aims to improve the detection of moving objects and enlarge per-pixel size of extracted features.
- **Object detection module:** Consists of a modified version of YOLOv4 detector [36], [53], customized to better detect small pedestrians recorded by a bird's eye view camera.
- **Multiple object tracking module:** Unlike street-level view, in bird's eye view, object occlusion barely occurs. Therefore, we can use a simpler tracker, SORT [54], which achieves sufficient accuracy and fast inference speed.
- **Group detection module:** Determines social groups and social distancing failures based on trajectory stability and pedestrians velocity similarities.

TABLE III: Quantitative Comparison on the B-SDA Dataset

WMBS	CC	AP	mIoU	Precision	Recall
		44.9	**71.7**	74.2	49.9
✓		55.1	69.8	70.9	62.9
	✓	58.0	68.75	**84.1**	62.8
✓	✓	**63.0**	68.77	73.3	**73.0**

A. Data Pre-Processing

The use of highly elevated cameras results in small and potentially blurry pedestrians. Videos with various lighting and weather conditions additionally impact the accuracy of object detection and tracking. To tackle these, we apply data pre-processing methods: Weighted-Mask Background Subtraction (WMBS) and Video Calibration (VC). WMBS constructs the background image from videos acquired by static cameras, computed as the mean of all N frames [55]. The background image with a weighted parameter α is subtracted from the original frames, to calculate the enhanced image. Formally,

$$F_b(I_r^{(t)}) = I_r^{(t)} - \frac{\alpha}{N} \sum_{k=1}^{N} I_r^{(k)}, \tag{1}$$

where $I_b^{(t)} = F_b(I_r^{(t)})$ represents the output image, $I_r^{(t)}$ is t-th frame in the original video, and α is the weight coefficient.

VC transforms bird's-eye view videos into calibrated bird's-eye videos perpendicular to the ground. It maps a trapezoidally distorted traffic intersection scene into a rectangular one with a uniform scale. Calibration is achieved by calculating the homography matrix M_{ca} that maps $I_b^{(t)}$ in image coordinates to $F_c(I_b^{(t)})$ in real world coordinates. Center cropping is the final stage in calibration, which removes unnecessary parts of the original image to increase the per-pixel size of features. The cropped image $I^{(t)}$ is the input for procedures that follow.

Table III shows how data pre-processing methods affect our customized YOLOv4 model. We select Weighted-Mask Background Subtraction (**WMBS**) and Center Cropping (**CC**). With these two methods, YOLOv4 achieves the highest AP and recall compared to other combinations (for crowd/traffic surveillance applications, recall is more important than precision). The MOT accuracy is evaluated by the CLEAR metrics [56], where MOTA is the key evaluation score. The tracking performance is evaluated on the B-SDA test dataset. The detection is generated by YOLOv4 with WMBS and Center Cropping. For the YOLOv4-SORT pipeline, we obtain

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3305587

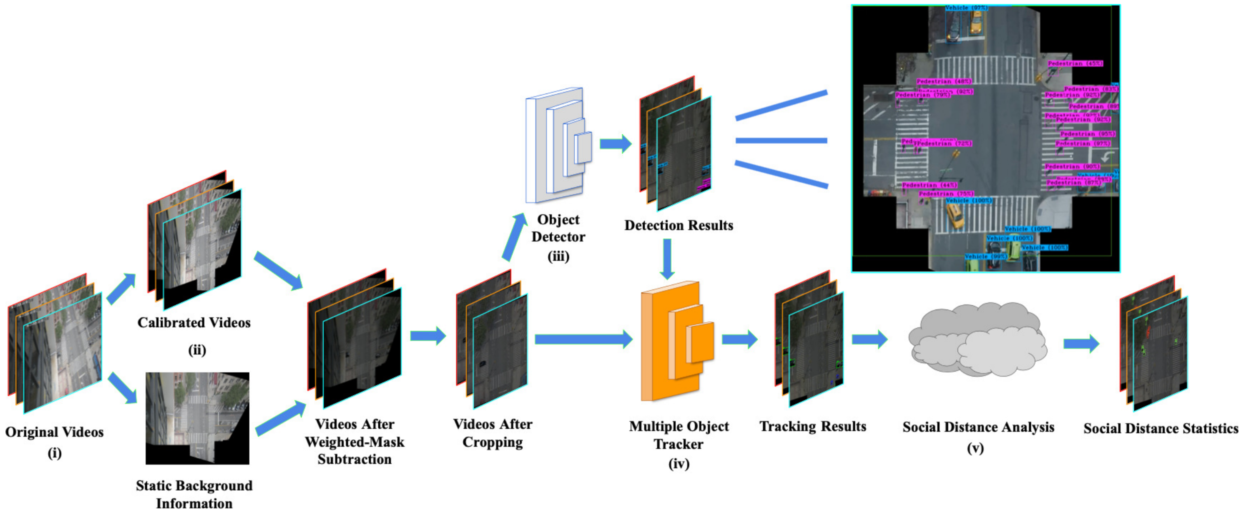

Fig. 7: Pipeline for the B-SDA system: (i) collect raw videos from a bird's-eye view camera; (ii) apply calibration and background subtraction to alleviate the effect of sub-optimal sensor quality; (iii) perform pedestrian detection; (iv) execute pedestrian tracking; (v) analyze the behavior of pedestrian movement using social distancing analysis algorithm.

TABLE IV: Annotation Statistics

Dataset	Number of Frames	Number of Objects
B-SDA train	7.4k	49.7k
B-SDA test	8.1k	203.2k

TABLE V: Group Validation Performance

Trajectory Compare	Velocity Compare	Precision	Recall	F1
		0.92	0.57	0.66
✓		0.90	**0.99**	**0.92**
✓	✓	0.86	0.96	0.88

MOTA = 47.65%, MOTP = 71.4%, MT = 60.9%, and ML = 5.8%.

B. Object Detection and Tracking

To reach the detection accuracy appropriate for the proposed social distancing analysis system, we altered the feature map topology in YOLOv4 to adopt a shallower feature map and to detect small pedestrians. The anchor sizes were determined based on the clustering results of the B-SDA dataset. In the training process of YOLOv4, the customized YOLOv4 started with the backbone pre-trained on the Imagenet dataset [57]. Next, it was trained with (a) VisDrone2019 dataset [58] in 832×832 resolution for 6,000 epochs, followed by (b) B-SDA dataset that consists of annotated videos recorded from the COSMOS camera (the annotation statistics are shown in Table IV) for another 6,000 epochs. We used a batch size of 64 and the learning rate of 10^{-3} with a weight decay of 5×10^{-4}. For real-time tracking, we use the SORT algorithm [59] which balances accuracy and processing speed.

C. Group Detection and Social Distancing Failure Detection

Similar to Auto-SDA, the social distancing analysis system in B-SDA continually receives the tracking information for each frame. The system keeps updating the tracking state and extracts useful information to obtain the pedestrians' trajectories.

Unlike Auto-SDA, the estimation of real-world distances between objects is simplified by the bird's-eye video calibration. The distance of six feet in our videos is represented by approximately 35 pixels based on the ground measurement.

Next, we create a Euclidean distance matrix for all detected pedestrians to find potential social distancing failure pairs. To avoid overcounting the number of failures, we use the group detection module described in Sec. III-D.

To evaluate the precision of the group detection module, we annotated groups of pedestrians with bounding boxes that cover all pedestrians within the same group, for $10,000$ video frames. In each group bounding box, pedestrians who are within the social distancing threshold are the true positives in the group validation evaluation. We use precision, recall, and F1 score for the evaluation.

Table V shows that the algorithm can capture accurate grouping information and filter out failure pairs belonging to the same group. We observe that trajectory Comparison significantly improves the Recall and F1 score, while velocity comparison has a negative impact on performance. As a result, we have decided to remove the velocity estimation function and rely solely on the trajectory comparison function for further analysis of social distancing.

V. Measurements and Evaluation

We applied Auto-SDA and B-SDA to videos recorded from the COSMOS cameras, which are deployed on the 2nd and 12th floor of Columbia's Mudd building looking at the COSMOS site (see Fig. 1).[3] Auto-SDA's dataset consists of 180 sec (two times the signal timing cycle of the traffic lights at the intersection) videos, recorded five times a day at 9 AM, 2 PM, 5:30 PM, 7:30 PM, and 10 PM, between June 17 and

[3] The use of the videos by Columbia researchers is IRB-exempt. The videos are solely used for research-related purposes. A data set of anonymized videos recorded by COSMOS cameras deployed on Columbia's Mudd building is available for researchers in [60]. The details of anonymization process are presented in [61].

July 2020 (after the lockdown), during May 2021 (after the vaccines became broadly available). It also includes 16 videos that were opportunistically recorded before the COVID-19 outbreak (in July 2019). B-SDA's dataset consists of 300 sec videos recorded at 9 AM, 2 PM, 5:30 PM, and 10 PM, from June 2020 to February 2021. It also includes sporadically collected videos between June and July 2019 (prior to the pandemic). We used the results to evaluate the impacts of the pandemic on pedestrians' social behavior.

Figures 8 to 13 present the results obtained by applying Auto-SDA on the recorded videos from the 2nd floor camera. Fig. 8 shows the fraction of recorded videos in which a certain percentage of pedestrians are walking as a group. One can see that the fraction of pedestrians walking as a social group has grown from 0-20% (during the lockdown) to 10-45% (post-vaccine). For each video, we calculated the percentage of pedestrians who neglect social distancing and plotted a normalized histogram of the results in Fig. 9. It can be seen that after the lockdown, less than 55% of the pedestrians neglected social distancing rules, compared to 65% post-vaccine. Fig. 10 compares the duration of social distancing failure incidents during the pandemic and post-vaccine. The results demonstrate a statistical increase of around 3 s in the duration of social distancing failures in the post-vaccine period. Fig. 11 displays the increase in the maximum duration of post-vaccine social distancing failure incidents. Fig. 12 illustrates the normalized histogram of the number of social distancing failures at different times of the day. We compare the pre-pandemic, lockdown, and post-vaccine density of the pedestrians at the intersection in Fig. 13. One can observe that density of the pedestrians has decreased by almost 50% after the lockdown (compared to pre-pandemic), while it has slightly increased after the availability of the vaccines.

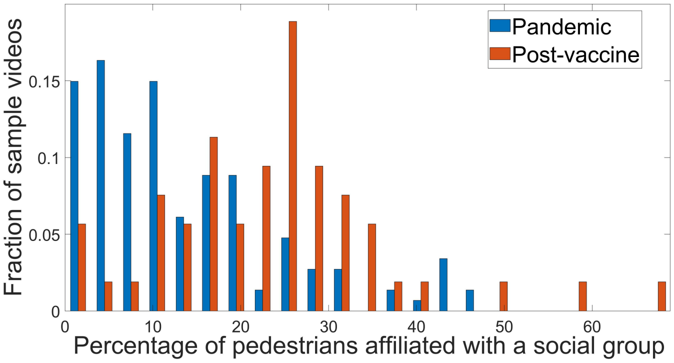

Fig. 8: Auto-SDA: Normalized histogram of the percentage of pedestrians affiliated with a social group.

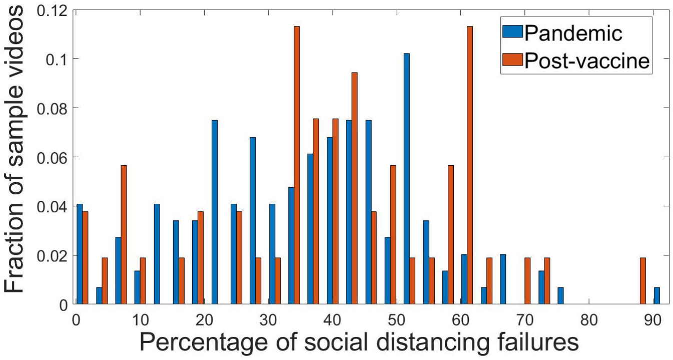

Fig. 9: Auto-SDA: Normalized histogram of the percentage of pedestrians who failed to comply with social distancing guidelines.

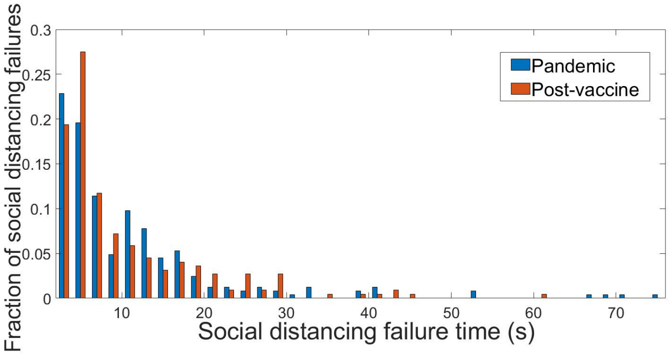

Fig. 10: Auto-SDA: Normalized histogram of the duration of the detected social distancing failure incidents.

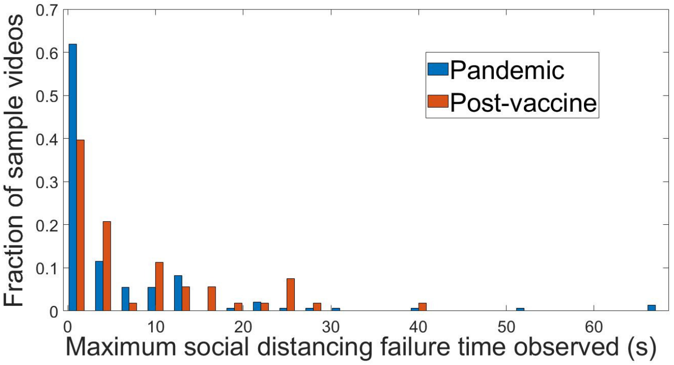

Fig. 11: Auto-SDA: Normalized histogram of the maximum duration of social distancing failure observed.

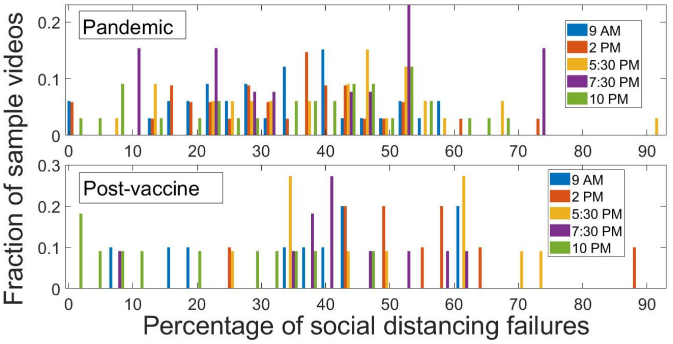

Fig. 12: Auto-SDA: Normalized histogram of the number of pedestrians neglecting social distancing protocols at different times of the day.

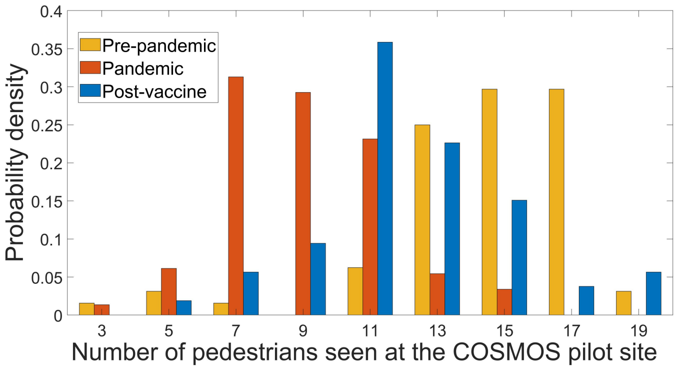

Fig. 13: Auto-SDA: Comparison between the density of pedestrians walking at the COSMOS site in different periods.

The results obtained by applying B-SDA on the videos recorded by the 12th floor camera are summarized in Figures 14 and 15. Specifically, Fig. 14 shows the probability distribution of the angle between the moving directions of two pedestrians in a failure pair before and during the pandemic, interpolated using Gaussian kernel density estimation. We declare that a face-to-face failure occurs when the difference in velocity direction is larger than 150 degrees. Before the pandemic, 42.3% of failures are face-to-face. During the pandemic, the distribution indicates that pedestrians are aware of higher chances of getting infected when neglecting social distancing. They are thus more cautious when walking towards each other, which decreases the percentage of face-to-face failures from 42.3% to 20.7%. We use a histogram to visualize the statistics of average per-minute failures at different times of day in Fig. 15. Considering that people are more likely to come into contact with each other when crowd density is high, it makes sense that the average number of failures is higher during the daytime.

Finally, we note that several surveys have been conducted across the U.S. e.g., [5], [10], [11], to measure individuals' compliance with social distancing guidelines during the peak of the COVID-19 pandemic in 2020 and following the widespread availability of vaccines in 2021. The results of these surveys indicate very similar trends in terms of adherence to social distancing policies and engagement in social interactions from the early stage of the outbreak to after the availability of the vaccines.

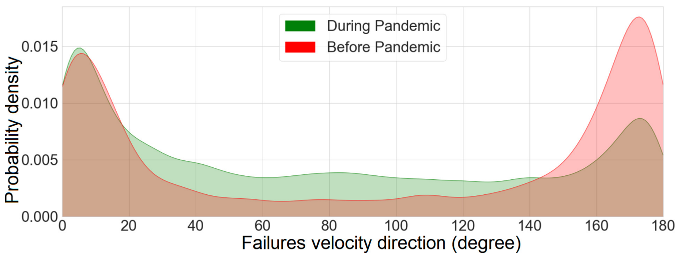

Fig. 14: B-SDA: Distribution of the angle between moving directions of two pedestrians in a failure pair.

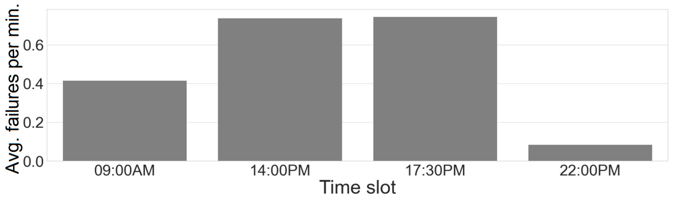

Fig. 15: B-SDA: Number of pedestrians who failed to comply with social distancing at different times of the day.

VI. Conclusion

We developed two approaches to measure compliance with social distancing: Auto-SDA and B-SDA. Auto-SDA uses street-level cameras and has high accuracy in calculating distances between pedestrians. B-SDA uses bird's eye view cameras to maintain privacy while still offering comparable accuracy. We applied Auto-SDA and B-SDA on videos from COSMOS cameras recorded before the pandemic, during the peak of the pandemic, and after the availability of the vaccines. The results represent the impacts of the social distancing rules on pedestrians' social behavior. The obtained results are consistent with the findings of conducted surveys in the U.S., showing fewer failures during the pandemic than before. After the vaccine was available, there was a slight increase in failures compared to the beginning of the outbreak.

VII. Future Work

Future research includes extending the proposed systems to integrate data from multiple cameras and other sensors such as LiDARs. Real-time operation of the systems is also an important aspect that must be considered, as this capability could facilitate immediate intervention and policy adjustments as necessary. These improvements will fully automate the evaluation of compliance with preventive measures such as social distancing policies and increase preparedness for future pandemics. The ultimate goal of these enhancements is to significantly boost the accuracy and speed of public health measures' effectiveness evaluations, thereby contributing to improved public safety and health outcomes.

Acknowledgments

We thank Ivan Seskar, Jakub Kolodziejski, and Michael Sherman (Rutgers/WINLAB).

References

[1] M. Ghasemi, Z. Kostic, J. Ghaderi, and G. Zussman, "Auto-SDA: Automated video-based social distancing analyzer," in *Proc. ACM HotEdgeVideo'21*, 2021.

[2] Z. Yang, M. Sun, H. Ye, Z. Xiong, G. Zussman, and Z. Kostic, "Bird's-eye view social distancing analysis system," in *Proc. IEEE ICC Workshop on Edge Learning for 5G Mobile Networks and Beyond*, 2022.

[3] M. Ghasemi, Z. Yang, M. Sun, H. Ye, Z. Xiong, J. Ghaderi, Z. Kostic, and G. Zussman, "Demo: Video-based social distancing evaluation in the COSMOS testbed pilot site," in *Proc. ACM MobiCom'21*, 2021.

[4] M. E. Kuiper, M. Chambon, A. L. de Bruijn, C. Reinders Folmer, E. H. Olthuis, M. Brownlee, E. B. Kooistra, A. Fine, F. van Harreveld, G. Lunansky *et al.*, "A network approach to compliance: A complexity science understanding of how rules shape behavior," *J. Bus. Ethics*, pp. 479–504.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3305587

[5] C. Vatovec and J. Hanley, “Survey of awareness, attitudes, and compliance with COVID-19 measures among Vermont residents,” *PloS One*, p. e0265014, 2022.

[6] E. M. Hoeben, W. Bernasco, L. Suonperä Liebst, C. Van Baak, and M. Rosenkrantz Lindegaard, “Social distancing compliance: A video observational analysis,” *PloS one*, p. e0248221, 2021.

[7] D. Raychaudhuri, I. Seskar, G. Zussman, T. Korakis, D. Kilper, T. Chen, J. Kolodziejski, M. Sherman, Z. Kostic, X. Gu, H. Krishnaswamy, S. Maheshwari, P. Skrimponis, and C. Gutterman, “Challenge: COSMOS: A city-scale programmable testbed for experimentation with advanced wireless,” in *Proc. ACM MobiCom’20*, 2020.

[8] S. Yang, E. Bailey, Z. Yang, J. Ostrometzky, G. Zussman, I. Seskar, and Z. Kostic, “COSMOS smart intersection: Edge compute and communications for bird’s eye object tracking,” in *Proc. SmartEdge’20*, 2020.

[9] Z. Duan, Z. Yang, R. Samoilenko, D. S. Oza, A. Jagadeesan, M. Sun, H. Ye, Z. Xiong, G. Zussman, and Z. Kostic, “Smart city traffic intersection: Impact of video quality and scene complexity on precision and inference,” in *Proc. IEEE Smart City’21*, 2021.

[10] A. C. Hansen, C. V. Farewell, J. S. Jewell, and J. A. Leiferman, “Exploring predictors of social distancing compliance in the United States during the COVID-19 pandemic,” *Disaster Med. Public Health Prep.*, p. e32, 2023.

[11] C. P. Reinders Folmer, M. A. Brownlee, A. D. Fine, E. B. Kooistra, M. E. Kuiper, E. H. Olthuis, A. L. de Bruijn, and B. Van Rooij, “Social distancing in America: Understanding long-term adherence to COVID-19 mitigation recommendations,” *PloS One*, p. e0257945, 2021.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Region-based convolutional networks for accurate object detection and segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, 2016.

[13] R. Girshick, “Fast R-CNN,” in *Proc. ICCV’15*, 2015.

[14] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.

[15] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” in *Proc. ICCV’17*, 2017.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” in *Proc. Eur. Conf. Computer Vision*, 2016.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proc. CVPR’16*, 2016.

[18] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in *Proc. CVPR’17*, 2017.

[19] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[20] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” *arXiv preprint:2004.10934*, 2020.

[21] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *Proc. IEEE ICIP’17*, 2017.

[22] S. Sun, N. Akhtar, H. Song, A. Mian, and M. Shah, “Deep affinity network for multiple object tracking,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 43, no. 1, pp. 104–119, 2019.

[23] 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, S. Chatzinotas, and B. Ottersten, “Enabling and emerging technologies for social distancing: A comprehensive survey,” *arXiv preprint arXiv:2005.02816*, 2020.

[24] A. Tyagi, D. Rajput, and A. Singh, “A review on social distancing auto detection techniques in perspective of COVID’19,” in *I-SMAC’21*, 2021.

[25] M. Gupta, M. Abdelsalam, and S. Mittal, “Enabling and enforcing social distancing measures using smart city and its infrastructures: A COVID-19 use case,” *arXiv preprint arXiv:2004.09246*, 2020.

[26] A. H. Ahamad, N. Zaini, and M. F. A. Latip, “Person detection for social distancing and safety violation alert based on segmented ROI,” in *Proc. IEEE ICCSCE’20*, 2020.

[27] 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,” *arXiv preprint arXiv:2007.03578*, 2020.

[28] S. Saponara, A. Elhanashi, and A. Gagliardi, “Implementing a real-time, AI-based, people detection and social distancing measuring system for COVID-19,” *J. Real-Time Image Proc.*, pp. 1937–1947, 2021.

[29] N. S. Punn, S. K. Sonbhadra, and S. Agarwal, “Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLOv3 and DeepSort techniques,” *arXiv preprint arXiv:2005.01385*, 2020.

[30] Z. Shao, G. Cheng, J. Ma, Z. Wang, J. Wang, and D. Li, “Real-time and accurate UAV pedestrian detection for social distancing monitoring in COVID-19 pandemic,” *IEEE trans. Multimedia*, vol. 24, pp. 2069–2083, 2021.

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

[32] D. Birla, “Social distancing AI,” https://github.com/deepak112/Social-Distancing-AI, 2020.

[33] J. Betancourt, “Social distancing analyser,” https://github.com/JohnBetaCode/Social-Distancing-Analyser, 2020.

[34] T. Farrand, “Social Distancing,” https://github.com/FarrandTom/social-distancing, 2020.

[35] Y. C. Hou, M. Z. Baharuddin, S. Yussof, and S. Dzulkifly, “Social distancing detection with deep learning model,” in *ICIMU’20*, 2020.

[36] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020.

[37] Y. Wang, Y. Li, and J. Zheng, “A camera calibration technique based on OpenCV,” in *Proc. 3rd Int. Conf. Inf. Sci. Interact. Sci.*, 2010.

[38] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[39] A. Ranganathan, “The levenberg-marquardt algorithm,” *Tut. on LM Alg.*, 2004.

[40] M. Jüngel, H. Mellmann, and M. Spranger, “Improving vision-based distance measurements using reference objects,” in *Robot Soccer World Cup*, 2007.

[41] P. Alizadeh, “Object distance measurement using a single camera for robotic applications,” Ph.D. dissertation, Laurentian University of Sudbury, 2015.

[42] P. Pias, “Object detection and distance measurement,” https://github.com/paul-pias/Object-Detection-and-Distance-Measurement, 2020.

[43] J. Nickolls, I. Buck, M. Garland, and K. Skadron, “Scalable parallel programming with CUDA,” *Queue*, vol. 6, no. 2, pp. 40–53, 2008.

[44] S. Inaba and Y. Aoki, “Conversational group detection based on social context using graph clustering algorithm,” in *Int. Conf. Signal-Image Technol. Internet-Based Sys. (SITIS)*, 2016.

[45] R. Mazzon, F. Poiesi, and A. Cavallaro, “Detection and tracking of groups in crowd,” in *IEEE Int. Conf. Adv. Video Signal Based Surveill.*, 2013.

[46] S. Vascon and L. Bazzani, “Group detection and tracking using sociological features,” in *Group and crowd behavior for computer vision*. Elsevier, 2017, pp. 29–66.

[47] F. Solera, S. Calderara, and R. Cucchiara, “Socially constrained structural learning for groups detection in crowd,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 5, pp. 995–1008, 2015.

[48] F. Solera, S. Calderara, and R. Cucchiara, “Structured learning for detection of social groups in crowd,” in *10th IEEE Int. Conf. on Adv. Video Signal Based Surv.*, 2013.

[49] E. T. Hall, *The hidden dimension*. Garden City, NY: Doubleday, 1966, vol. 609.

[50] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T. Kim, “Multiple object tracking: A literature review,” *Artificial Intelligence*, vol. 293, p. 103448, 2014.

[51] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, “Tracking without bells and whistles,” in *Proc. ICCV*, 2019.

[52] S. Sun, N. Akhtar, H. Song, A. Mian, and M. Shah, “Deep affinity network for multiple object tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 104–119, 2018.

[53] Z. Yang, M. Sun, H. Ye, Z. Xiong, G. Zussman, and Z. Kostic, “Birds eye view social distancing analysis system,” *arXiv preprint:2112.07159*, 2021.

[54] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in *Proc. IEEE ICIP’16*, 2016.

[55] C. COW, “Averaging video frames,” https://www.youtube.com/watch?v=ZS1MbjyUNto, accessed February 3, 2022.

[56] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: the clear mot metrics,” *EURASIP J. Image Video Process*, vol. 2008, pp. 1–10, 2008.

[57] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[58] P. Zhu, L. Wen, D. Du, X. Bian, Q. Hu, and H. Ling, “Vision meets drones: Past, present and future,” *arXiv preprint:2001.06303*, 2020.

[59] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” *Proc. ICIP’16*, 2016.

[60] COSMOS Project, “Hardware: Cameras,” 2022. [Online]. Available: https://wiki.cosmos-lab.org/wiki/Hardware/Cameras.

[61] A. Angus, D. Z, G. Zussman, and Z. Kostic, “Real-time video anonymization in smart city intersections,” in *Proc. IEEE MASS’22*, 2022.