Bird's-eye View Social Distancing Analysis System

Zhengye Yang⋆, Mingfei Sun†, Hongzhe Ye†, Zihao Xiong†, Gil Zussman†, Zoran Kostic†
⋆Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY
† Department of Electrical Engineering, Columbia University, New York, NY
{yangz15}@rpi.edu,{ms5898, hy2610, zx2273, gil.zussman, zk2172}@columbia.edu

***Abstract*—Social distancing can reduce the infection rates in respiratory pandemics such as COVID-19. Traffic intersections are particularly suitable for monitoring and evaluation of social distancing behavior in metropolises. Hence, in this paper, we propose and evaluate a real-time privacy-preserving social distancing analysis system (B-SDA), which uses bird's-eye view video recordings of pedestrians who cross traffic intersections. We devise algorithms for video pre-processing, object detection, and tracking which are rooted in the known computer-vision and deep learning techniques, but modified to address the problem of detecting very small objects/pedestrians captured by a highly elevated camera. We propose a method for incorporating pedestrian grouping for detection of social distancing violations, which achieves 0.92 F1 score. B-SDA is used to compare pedestrian behavior in pre-pandemic and during-pandemic videos in uptown Manhattan, showing that the social distancing violation rate of 15.6% during the pandemic is notably lower than 31.4% pre-pandemic baseline.**

***Keywords*—Social distancing, Object detection, Smart city, Testbeds**

I. Introduction

Respiratory viruses such as COVID-19 are spread by individuals who are in close proximity to each other for a given period of time. Per CDC policies, individuals should maintain a social distance of at least 6 feet to suppress the spread of the virus [1]–[3]. Streets and traffic intersections are locations where social-distancing violations are prone to occur. It is desirable to provide precise measurement of social distancing in cities with a possible use case of providing data to aid epidemiological research. Furthermore, it is important to develop this technology with privacy preservation in mind.

Traffic intersections in metropolises are suitable for deployment of smart-city sensors, high-speed communications and edge computing nodes, which allow to collect, process and analyze high-bandwidth data such as videos used for monitoring of social distancing behavior [4], [5]. The results of this paper are based on the experiments performed on the NSF PAWR COSMOS testbed deployed in New York City [6], which contains sensor, communications and computing infrastructure which can specifically support privacy preserving social distancing analysis.

In traffic intersections, it is common to deploy cameras at low altitudes. This approach has several drawbacks: (i) limited view of individual cameras, (ii) further-away objects are occluded by closer objects, (iii) tracking of pedestrians is challenging due to occlusions, and (iv) face and license plate recognition can violate privacy.

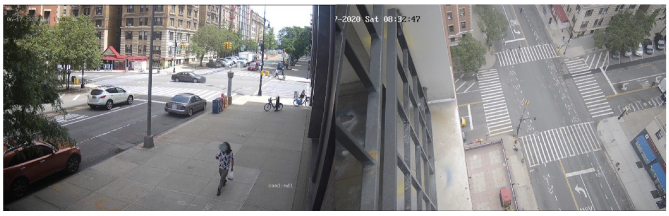

Fig. 1. Left: View from a low-elevation camera; Right: bird's-eye view from a high-elevation camera.

These challenges can be addressed by using cameras installed at high elevations, which record bird's-eye view videos as in Fig. 1(right). 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 [7]–[15], and (ii) Reliable tracking of pedestrian trajectories across video frames [16]–[19]. 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. For this paper, we acquired a large dataset of bird's-eye view videos, and annotated it for detection of pedestrians. We customized the object detection method YOLOv4 [15] and the multiple object tracking method SORT [20], such that they are able to provide a desirable inference speed and to achieve promising detection and tracking performance for social distancing applications.

A number of social distancing analysis prototypes have been recently proposed [21]–[23] (see also Section II). They are based on open-source datasets with low-altitude camera views, leading to potential privacy violations. Most utilize only the detection information to analyze social distancing, which is insufficient for acquisition of statistics such as pedestrian throughput at traffic intersections. When using only the distance between pedestrians to assess proximity violations, it is impossible to discriminate between "safe social groups" and random people in close proximity to each other. A "safe social group" is defined as a collection of people assumed to reside together, such as a family. Identifying "safe social groups" requires group tracking, in order not to declare the participating pedestrians as violators of social distancing.

Accordingly, in this paper, we propose and evaluate a bird's-eye social distancing analysis (B-SDA) system. The main contributions, as outlined in the following sections, are: (i)

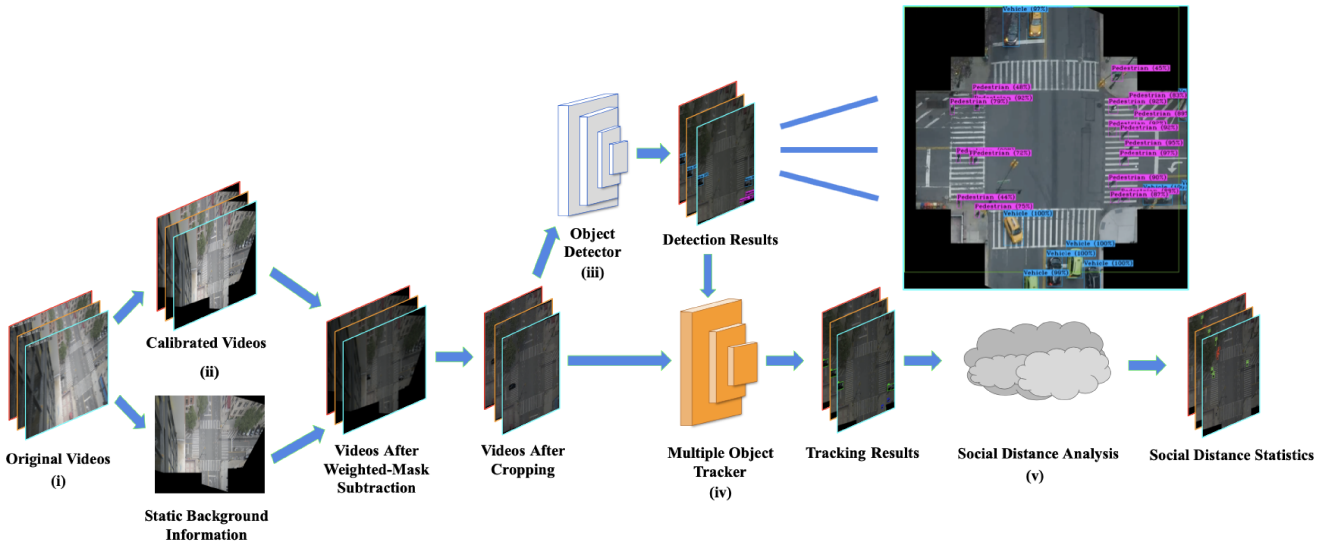

Fig. 2. Pipeline for the B-SDA system: (i) collect raw videos from a bird's-eye view camera; (ii) conduct calibration and background subtraction to alleviate the effect of sub-optimal sensor quality; (iii) obtain pedestrian detection results; (iv) obtain pedestrian tracking information; (v) analyze pedestrian movement behaviour with social distancing analysis algorithm.

design of a social distancing analysis system based on bird's-eye view videos (Fig. 2), (ii) design of social group identity validation algorithm, and (iii) comparison of social distancing behavior before and during the COVID-19 pandemic, for a major metropolis. Due to space constraints, some results were omitted and can be found in [24].

II. Related Work

A. Object Detection

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 [7], Fast R-CNN [8], Faster R-CNN [9] and Mask R-CNN [10] use a two-stage structure for object detection, which consists of region proposal stage and classification stage. In contrast, SSD [11] and YOLO methods [12]–[15] have a single-stage structure, achieving higher inference speeds. As more capable deep-learning backbone networks are designed and better data augmentation methods are devised, single-stage models evolve into many variants. Improvements of YOLO [13]–[15] make it possible to approach the detection accuracy comparable to R-CNN without sacrificing YOLO's inference speed in our traffic scenario. Considering the speed-accuracy trade-off, we chose YOLOv4 [15] as the baseline for object detection in this work.

B. Multiple Object Tracking

The task of Multiple Object Tracking (MOT) is partitioned into locating multiple objects, maintaining their identities, and obtaining individual trajectories given an input video [16]. SORT [20] is a tracking-by-detection algorithm whose aim is to assign a common tracking ID for detections of the same object in subsequent frames in a video. It performs imperfectly when tracking through occlusions. DeepSORT [18] algorithm replaces the association metric with a more informed one which combines motion and appearance information by using a convolutional neural network. Tracktor method [17] accomplishes multi-object tracking with the Faster R-CNN object detector only, but it underperforms when objects are moving rapidly or when the density of objects is high. Bird's eye view significantly reduces object occlusions in a scene, where visual features for re-identification are less useful for improving the tracking performance. We chose SORT [20] as the tracking module due to its simplicity and fast inference.

C. Social Distancing Analysis

Key aspects of the visual social distancing problem have been proposed in [25]. With the sudden outbreak of the COVID pandemic, the proposals for social distancing surveillance have been rapidly emerging. Given the nature of the social distancing problem, the primary task is to detect pedestrians and measure the distance between two individuals. In early works, the solutions were based on object detection with distance approximation [21], [23]. Mainstream object detection methods such as Faster R-CNN [9], Mask R-CNN [10], YOLOv3 [14] and YOLOv4 [15] are widely used. To collect more useful identity-related statistics, tracking-by-detection MOT method has been added into the workflow by [26], [27]. Considering the accuracy-speed tradeoff in tracking algorithms, efficient methods like SORT and DeepSORT are widely used for social distancing problems. Group information for crowds is further added to reduce false positives caused by naïvely chosen distance thresholds, and several methods for group detection have been recently proposed [28]–[32]. Most research contributions discussed above utilize street-level cameras that could violate pedestrian privacy [33]. On the other hand, bird's-eye view cameras, used in our research, provide

an alternative approach which achieves privacy preservation and provides a much larger surveillance area per camera.

D. Group Detection

Due to the inherent social nature of human behavior, interactions typically happen between small subsets of people referred to as groups [34]. Hall proposed the proxemic theory which defined distances between two people with different intimacy levels in the North American culture [35]. Many researchers investigated measurement metrics for group identification [34], [36]–[38]. Crowd understanding, or crowd analysis, is a topic closely related to group detection [39], [40]. Previous group detection/analysis approaches are unsuitable for our use case, which motivated us to design a new method.

III. Description of the B-SDA System

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 [41]. The background image with a weighted parameter α is subtracted from the original frames, to calculate the enhanced image

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

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.

B. Detection and Tracking

Object detection provides object localization and classification information to a Multiple Object Tracking (MOT) algorithm in a tracking-by-detection scheme. The detector provides $m^{(t)}$ proposed bounding boxes $D^{(t)} = \{d_1^{(t)}, d_2^{(t)}, ..., d_{m^{(t)}}^{(t)}\}$ from $I^{(t)}$, and $d_j^{(t)} = (dclass_j^{(t)}, bbox_j^{(t)}, conf_j^{(t)})$ records the class, location and confidence score of the j-th predicted box.

Once the MOT algorithm receives $D^{(t)}$, an identity association is performed to get the tracking state $S^{(t)} = \{s_1^{(t)}, s_2^{(t)}, ..., s_{m^{(t)}}^{(t)}\}$ for all $m^{(t)}$ objects in the t-th frame. $s_j^{(t)} = (class_j^{(t)}, roi_j^{(t)}, id_j^{(t)})$ denotes the state of the j-th object in the t-th frame, $class_j^{(t)}$ is the class of that object, $roi_j^{(t)}$ is the location of that object, and $id_j^{(t)}$ indicates the unique ID for that object.

C. Social distancing analysis

The social distancing analysis system continually receives the tracking information $S^{(t)}$ for each frame. The system keeps updating the tracking state S and extracts useful information to create the state history $T = \{T^{(1)}, T^{(2)}, ..., T^{(t)}\}$.

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 our ground measurement.

Next we create an Euclidean distance matrix for all objects in $T^{(t)}$ to find potential social distancing violation pairs $L^{(t)} = \{(bid_{11}^{(t)}, bid_{12}^{(t)}), ..., (bid_{p1}^{(t)}, bid_{p2}^{(t)})\}$, where $\{bid_{n1}^{(t)}, bid_{n2}^{(t)}\}$ denotes the n-th violation object ID pairs of all p pairs.

To avoid over-counting the number of violations, we design a cascade-condition filter to validate if a social distancing violation pair is (incorrectly) indicated. The assumption is that people belonging to the same "safe social group" are going to maintain the proxemic relationship while crossing the traffic intersection. The proxemic relationship can be captured by pedestrian trajectory information, which can be decomposed into three components: velocity similarity, trajectory similarity [42] and proxemic stability. The trajectory similarity is measured by the Euclidean distance between two pedestrians at the same temporal location, as shown in (2). The first order derivative of a trajectory can be used as the velocity estimator

$$sim(a,b) = \frac{\sum_{i \in X_a \bigcap X_b} \|(a,b)\|_2}{|X_a \bigcap X_b|} \quad . \tag{2}$$

The estimation of object velocity depends heavily on the correct localization of bounding boxes. Caused by imperfections of the object detection algorithm, oscillations in localization could seriously disturb the estimation of object velocities. To remedy this, we calculate velocities with exponentially weighted averages where velocities are recalculated based on weighted previous and current velocities. To compare the velocity similarity, both magnitude and direction need to be evaluated. We use cosine distance (D_{cos}) to measure the similarity in vector direction. In order to combine the magnitude similarity with the cosine distance measure, it needs to be scaled. We use the magnitude similarity (D_{Mag}) in (3) which is inspired by the formula discussed by Rima *et al.* [36].

$$D_{Mag} = \frac{|\,\|\,v_1\,\| - \|\,v_2\,\|\,|}{\max(\|\,v_1\,\|, \|\,v_2\,\|)} \tag{3}$$

The overall velocity distance $D(v1, v2)$ with weight parameter γ is described in (4). The ratio between the standard deviation of distance and the trajectory similarity between two violation candidates captures the stability of the proxemic relationship f_{stab}. The function Trajectory Compare is the "OR" condition filter between trajectory similarity and trajectory stability, which aims to be suitable for different group scenarios in a traffic intersection.

$$D(v1, v2) = \gamma \cdot D_{cos} + (1-\gamma) \cdot D_{Mag} \tag{4}$$

TABLE I
ANNOTATION STATISTICS

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

TABLE II
RECORDING SCHEDULE AND CORRESPONDING STATISTICS DURING THE PANDEMIC

Recording Schedule	Number of Frames
09:00-09:05	137.7k
14:00-14:05	140.4k
17:30-17:35	148.5k
22:00-22:05	143.1k

TABLE III
GROUP VALIDATION PERFORMANCE

Traj. Comp.	Vel. Comp.	Precision	Recall	F1
		0.92	0.57	0.66
✓		0.90	**0.99**	**0.92**
✓	✓	0.86	0.96	0.88

TABLE IV
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**

If the velocity similarity between two objects lies within a threshold, the objects are evaluated by the trajectory comparison. If all comparisons pass the test, the two objects are declared to belong to the same group, and removed from the social distancing violation list $L^{(t)}$. After these modifications, the list can be used to provide an index to collect the statistics.

IV. EXPERIMENTS

A. Data Acquisition

The training dataset is composed from two resources: (i) Public dataset Visdrone2019 [43], and (ii) Traffic intersection videos recorded in New York City, and annotated by our group (B-SDA dataset). The videos were recorded by the Hikvision (DS-2CD5585G0-IZHS) camera at the rate of 15 frames per second with 1920×1080 resolution. The annotation statistics are shown in Table I. For inference purposes, the videos were recorded multiple times per day from June 2020 to Febuary 2021 (during the pandemic and before broad availability of vaccines), with the recording schedule and corresponding statistics shown in Table II.

B. Experimental Setup

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 [44]. Next, it was trained with (a) VisDrone2019 dataset [43] in 832×832 resolution for 6,000 epochs, followed by (b) B-SDA dataset 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 tracking, we use the SORT algorithm [20] for real-time processing without sacrificing much in accuracy. In the group validation algorithm, we use D_{cos}, D_{Mag} and $D(v1, v2)$ to check the velocity similarity. Parameters γ and λ are set to 0.1 and 1 respectively. The threshold for velocity similarity is set to 0.21. For similarity measurement for trajectories, we use (2) and f_{stab}. The social distance threshold for (2) is equal to 35. The trajectory stability threshold for f_{stab} is set to 0.25. The computational setup comprises of the operating system Ubuntu 18.04 running on a cluster of 8 vCPUs, 30GB RAM, and one Tesla P100 GPU.

C. Verification of the Group Validation Algorithm

We evaluate the effectiveness of our algorithm for validation of pedestrian groups. We annotate groups of pedestrians with bounding boxes that cover all pedestrians within a same group, for $10,000$ video frames. In each group bounding box, pedestrians who are within the social distancing threshold (35 pixels) are the true positives in the group validation evaluation. We use precision, recall and F1 score for the evaluation.

Table III shows that our algorithm can capture accurate grouping information, and filter out violation pairs in the same group (Traj. Comp. denotes Trajectory Compare and Vel. Comp. denotes Velocity Compare). We observe that Trajectory Comparison brings significant improvement to the Recall and to the F1 score, whereas Velocity Comparison is harmful to the performance. We postulate that velocity estimation introduces large amount of noise even when equipped with the exponentially weighted average. Therefore, we remove the function Velocity Estimation, and use only the function Trajectory Comparison for further analysis of social distancing.

V. RESULTS

A. Detection and Tracking for Bird's-Eye View Videos

Precise object detection and tracking are essential to our social distancing system. Table IV shows how data preprocessing methods affect our customized YOLOv4 model. We select Weighted-Mask Background Subtraction (**WMBS**) and Center Cropping (**CC**). For safety-critical traffic surveillance, recall is more important than precision. The MOT accuracy is evaluated by the CLEAR metrics [45], 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 MOTA = 47.65%, MOTP = 71.4%, MT = 60.9%, and ML = 5.8%. The pipeline provides accurate object locations and identity and the group validation provides reliable social distancing analysis.

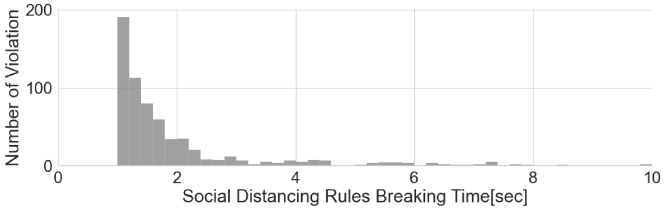

Fig. 3. Distribution of duration of social distancing violations.

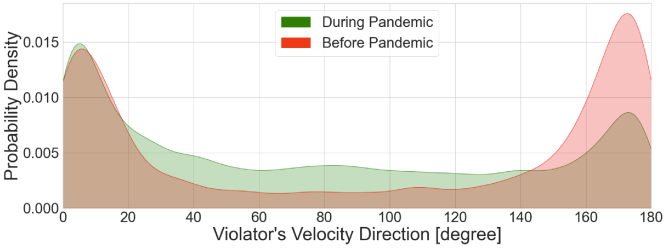

Fig. 4. Distribution of the angle between moving directions of two pedestrians in a violation pair.

B. Social Distancing Analysis

We collected the statistical information about the time duration of violations for all social distancing violators in 458 videos recorded during the COVID-19 pandemic (and before vaccines became widely available) in New York City. For comparison, we also performed the analysis on another B-SDA video dataset which was collected between June and July 2019 (prior to the pandemic). To simplify the evaluation and visualization of the statistics, we note that walking pedestrians may not be able to maintain the social distance of exactly 6 feet when crossing a street, which will cause violations that are less than one second in duration. These short violations would count pedestrians who behave properly as social distancing violators. Therefore, in the following analysis and figures we omit all violations which are shorter than 1 second. The estimation of pedestrian density (number of pedestrians) relies on the number of trajectories. ID switches caused by imperfect tracking enlarge the estimated number of pedestrians. To better estimate the pedestrian density, the true average trajectory length can be used as the reference. From the annotated video recorded during the pandemic, the real world average pedestrian trajectory length is 19.11 seconds. In the 458 recordings, the average pedestrian trajectory length extracted from tracking inference is 5.63 seconds. This difference reflects the fact that the ID switches enlarge the perceived number of pedestrians. To alleviate this effect, we calculate $ED = TC \cdot \frac{AT_{\text{GT}}}{AT_{\text{Infer}}}$ to estimate the actual pedestrian density, where TC is the total number of trajectories, AT_{GT} is the average trajectory length from the ground truth, and AT_{Infer} is the average trajectory length obtained from inference results.

Fig. 3 shows the histogram of the duration of violations, where 75 percent of violations are shorter than 4 seconds.

Fig. 4 shows the probability distribution of the angle between moving directions of two pedestrians in a violation pair before and during the pandemic, estimated by KDE with Gaussian kernel. We define that a face-to-face violation occurs when the difference in velocity direction is larger than 150 degrees. Before the pandemic, 42.3% of violations are face-to-face. During the pandemic, the distribution clearly indicates that pedestrians are aware of higher chances of getting infected when violating social distancing, and are thus more cautious when walking towards each other, which decreases the percentage of face-to-face violations from 42.3% to 20.7%.

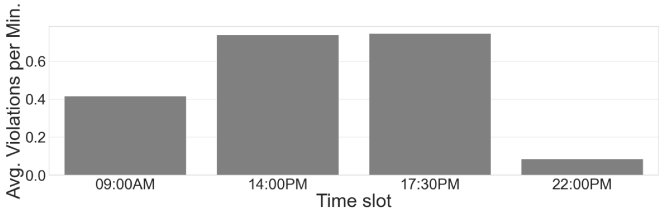

Fig. 5. Number of pedestrians who violate social distancing at different times of day.

We use a histogram to visualize the statistics of average per minute violations, at different times of day, in Fig. 5. 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 violations is higher during the day time. We also compare the proportion of social distancing violations between 2020-1 (during the pandemic) and 2019 (before the pandemic). If we evaluated 2019 data using 2020-1 (pandemic) social distancing rules, 31.4% of people would be judged as rule violators. Using the same criteria, only 15.6% of people are considered as rule violators during the 2020-1 period. During the pandemic people are aware of social distancing rules, and they deliberately keep distance from each other. Additionally, lockdown requirements reduced the density of pedestrians on the streets, which provided more space for pedestrians to deliberately perform social distancing.

C. The Speed of Inference

We record the inference speed of individual model components. Note that FP16 TensorRT implementation (30.9 FPS) of YOLOv4 is three times faster than FP32 implementation (11.1 FPS), without observable loss in object detection accuracy. SORT algorithm (553 FPS) and social distancing analysis (878 FPS) are almost negligible compared to YOLOv4. The overall system achieved 28.3 FPS (FP16) and 10.7 (FP32).

VI. Conclusion

We developed B-SDA, a privacy-preserving system for measurement and analysis of social distancing behavior in traffic intersections based on bird's-eye view video recordings, which incorporates a group validation technique to eliminate false positives in detecting social distancing violations. We collected, conditioned and annotated a dataset of several hundred videos of an intersection in New York City before and during the COVID-19 pandemic (a demonstration of the results was conducted in [46]) The analyzed results were presented as probability functions which capture the type and density of social distancing violations at different times of day and under a variety of conditions. The obtained quantitative

results correlate well with anticipated and visually observed pedestrian behavior, most dramatically represented by the fact that social distancing violations are estimated to be much less frequent during the pandemic (15.6%) when compared to pre-pandemic (31.4%). The proposed technique can be used as an epidemiological tool for managing respiratory pandemics.

ACKNOWLEDGEMENT

This work was supported in part by NSF grants CNS-1827923, OAC-2029295, CNS-2038984, and AT&T VURI award.

REFERENCES

[1] L. Matrajt and T. Leung, "Evaluating the effectiveness of social distancing interventions to delay or flatten the epidemic curve of coronavirus disease," *Emerging infectious diseases*, vol. 26, no. 8, p. 1740, 2020.

[2] Centers for Disease Control and Prevention, "Social distancing," https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/social-distancing.html, accessed February 3, 2022.

[3] ——, "Contact tracer's interview tool: Notifying people about an exposure to COVID-19," https://www.cdc.gov/coronavirus/2019-ncov/php/notification-of-exposure.html, accessed March 15, 2022.

[4] 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.

[5] 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.

[6] 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*, 2020.

[7] 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.

[8] R. Girshick, "Fast R-CNN," in *Proc. ICCV'15*, 2015.

[9] 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.

[10] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. ICCV'17*, 2017.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. ECCV'16*, 2016.

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

[13] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. CVPR'17*, 2017.

[14] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint:1804.02767*, 2018.

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

[16] 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.

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

[18] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. ICIP'17*, 2017.

[19] 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.

[20] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," *Proc. ICIP'16*, 2016.

[21] D. Birla, "Social-distancing-ai," https://github.com/deepak112/Social-Distancing-AI/tree/08a9a21ccf8ced3e6ff270628cb1c9b21a55fbee, accessed February 3, 2022.

[22] N. Singh 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:2005.01385*, 2020.

[23] 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:2007.03578*, 2020.

[24] 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.

[25] M. Cristani, A. Del Bue, V. Murino, F. Setti, and A. Vinciarelli, "The visual social distancing problem," *IEEE Access*, vol. 8, pp. 126 876–126 886, 2020.

[26] 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:2005.01385*, 2020.

[27] S. Gupta, R. Kapil, G. Kanahasabai, S. S. Joshi, and A. S. Joshi, "SD-Measure: A social distancing detector," in *Proc. IEEE 12th Int. Conf. on Computational Intelligence and Communication Networks (CICN)*, 2020.

[28] M. Rezaei and M. Azarmi, "Deepsocial: Social distancing monitoring and infection risk assessment in COVID-19 pandemic," *Applied Sciences*, vol. 10, no. 21, p. 7514, 2020.

[29] P. Sun, G. Draughon, and J. Lynch, "An autonomous approach to measure social distances and hygienic practices during COVID-19 pandemic in public open spaces," *arXiv preprint:2011.07375*, 2020.

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

[31] F. Class-Peters, W. Y. H. Adoni, T. Nahhal, A. E. Byed, M. Krichen, C. Kimpolo, and F. M. Kalala, "Post-COVID-19: Deep image processing AI to analyze socialdistancing in a human community," in *Adv. Smart Soft Comput.* Springer Singapore, 2022, vol. 1399.

[32] T. Chowdhury, A. Bhatti, I. Mandel, T. Ehsan, W. Ju, and J. Ortiz, "Towards sensing urban-scale COVID-19 policy compliance in new york city," in *Proc. ACM BuildSys'21*, 2021.

[33] B. Benfold and I. Reid, "Stable multi-target tracking in real-time surveillance video," in *Proc. CVPR'11*, 2011.

[34] F. Solera, S. Calderara, and R. Cucchiara, "Structured learning for detection of social groups in crowd," in *10th IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*, 2013.

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

[36] R. Chaker, Z. Al Aghbari, and I. N. Junejo, "Social network model for crowd anomaly detection and localization," *Pattern Recognition*, vol. 61, pp. 266–281, 2017.

[37] W. Ge, R. T. Collins, and R. B. Ruback, "Vision-based analysis of small groups in pedestrian crowds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, p. 1003–1016, May 2012.

[38] M. Ehsanpour, A. Abedin, F. Saleh, J. Shi, I. Reid, and H. Rezatofighi, "Joint learning of social groups, individuals action and sub-group activities in videos," *arXiv preprint:2007.02632*, 2020.

[39] N. Liu, Y. Long, C. Zou, Q. Niu, L. Pan, and H. Wu, "ADCrowdNet: An attention-injective deformable convolutional network for crowd understanding," *arXiv preprint:1811.11968*, 2018.

[40] Y. Liu, M. Shi, Q. Zhao, and X. Wang, "Point in, box out: Beyond counting persons in crowds," in *Proc. CVPR'19*, 2019.

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

[42] N. Magdy, M. A. Sakr, T. Mostafa, and K. El-Bahnasy, "Review on trajectory similarity measures," in *IEEE 7-th Int. Conf. on Intelligent Computing and Information Systems (ICICIS)*, 2015.

[43] 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.

[44] 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.

[45] 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.

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