Auto-SDA: Automated Video-based Social Distancing Analyzer

Mahshid Ghasemi, Zoran Kostic, Javad Ghaderi, Gil Zussman
Department of Electrical Engineering, Columbia University
{mahshid.ghasemi,zk2172,jghaderi,gil.zussman}@columbia.edu

wim.net
Wireless & Mobile Networking Lab

COSMOS

NEAL
Network Algorithms

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

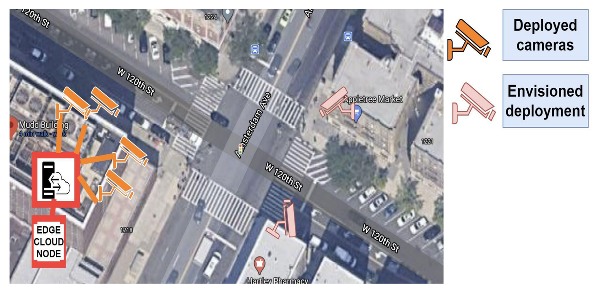
Abstract

Social distancing can reduce infection rates in respiratory pandemics such as COVID-19, especially in dense urban areas. To assess pedestrians' compliance with social distancing policies, we use the pilot site of the PAWR COSMOS wireless edge-cloud testbed in New York City to design and evaluate an **Auto**mated video-based **S**ocial **D**istancing **A**nalyzer (**Auto-SDA**) pipeline.

- Auto-SDA uses the state-of-the-art YOLOv4 object detector and the Nvidia DCF-based tracker to derive pedestrians' trajectories and measure the duration of proximity events.
- We incorporate 3 modules into Auto-SDA:
  - Calibration module that converts 2D pixel distances to 3D on-ground distances with less than 10 cm error.
  - Correction module that increases the accuracy of the tracker and object detector model.
  - Group detection module that identifies the pedestrians who walk together as a social group and excludes them from the social distancing violation analysis.
- We applied Auto-SDA to videos recorded at the COSMOS pilot in 3 periods of time, i.e., pre-pandemic, soon after the lockdown (pandemic), and after the vaccines became broadly available (post-pandemic), and analyzed the impacts of the social distancing protocols on pedestrians' behaviors and their evolution.

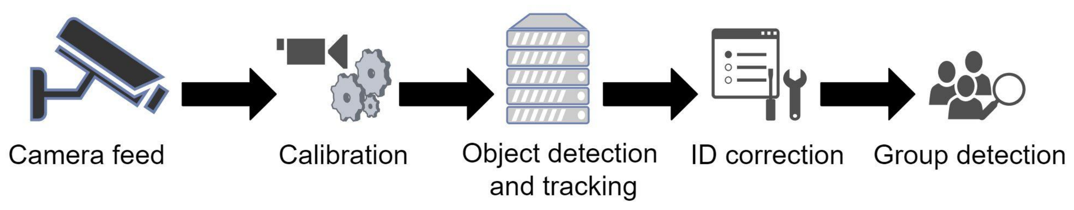
COSMOS-Testbed

- COSMOS architecture has a particular focus on ultra-high bandwidth and low latency wireless communication tightly coupled with edge cloud computing.
- We used this infrastructure to build a video analytic pipeline (**Auto-SDA**) that evaluates the compliance of pedestrians with the social distancing policies.



Cameras are deployed on the Columbia University Mudd building and connected to the edge-cloud servers via dedicated fibers.

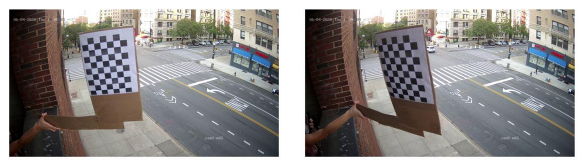
Auto-SDA Pipeline



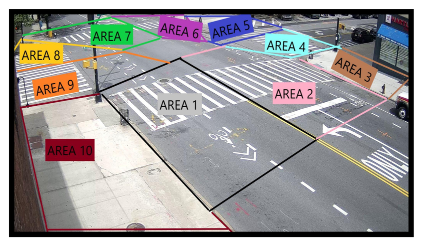
- **Calibration module** determines the intrinsic and extrinsic parameters of the camera to convert the 2D on-image coordinates, viewed by the camera, to the 3D on-ground coordinates.
- **Object detection and tracking module** locates the pedestrians and assigns an ID to each of them.
- **ID correction module** removes the redundant IDs generated by the tracker.
- **Group detection module** excludes the pedestrians affiliated with one social group from social distancing violation.

Calibration Module

- Intrinsic and extrinsic parameters of the cameras are required to convert pixel distances to on-ground distances.
- Intrinsic parameters consist of:
  Principal point; Focal length in pixel units; Radial distortion coefficients; Tangential distortion coefficients.
  - Extrinsic parameters consist of:
    Rotation vector R; Translation vector t.
- 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.



Calibration of the COSMOS cameras using a checkerboard.



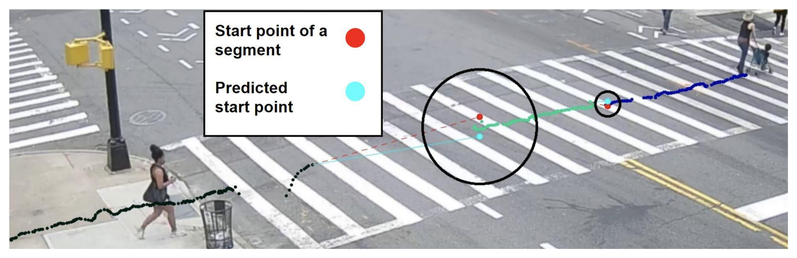
The camera scene is divided to 10 areas and the extrinsic parameters are calculated for each area individually.

ID Correction

- ID correction 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.
- If multiple IDs are assigned to a single pedestrian, this module removes the redundant IDs.
- The corrected IDs are used to derive the entire trajectory of that pedestrian.

Algorithm 1 ID Correction

Input: $ID_{\text{vec}}, e_1, e_2, n$ ▷ ID_{vec} is the output of NvDCF tracker
Output: corrected ID_{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 $id.Trj.head(n)$
for $(id_1, id_2) \in ID_{\text{vec}}$ **do**
  $t1 \leftarrow id_1.TimeStamp.StopTime$
  $t2 \leftarrow id_2.TimeStamp.StartTime$
  $p_1 \leftarrow id_1.TailEst.\text{at}(t = t_2),\ \ p_2 \leftarrow id_1.Trj.\text{at}(t_2)$
  $v_1 \leftarrow id_1.TailDir,\ \ v_2 \leftarrow id_2.HeadDir$
  if $t_2 - t_1 < e_1$ && $|p_1 - p_2| < e_2$ && $\angle(v1, v2) < 90^\circ$ **then**
    id_1 and id_2 belongs to same person



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

Group Detection

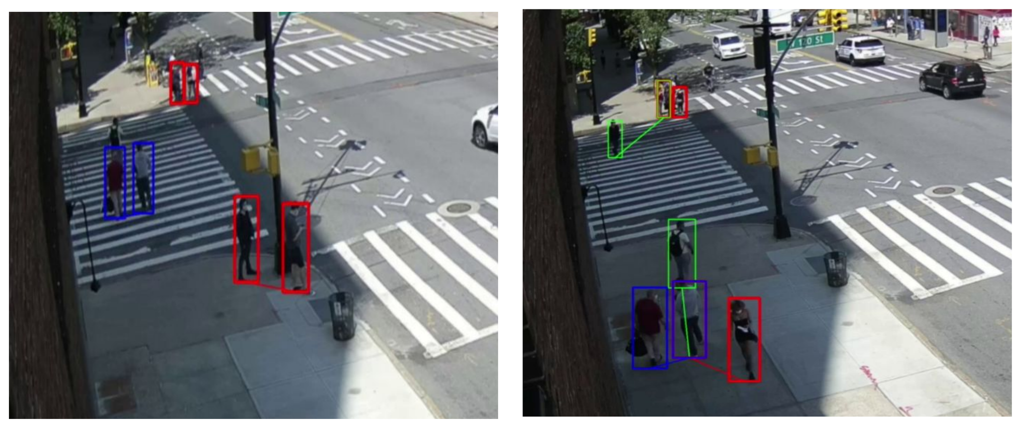
- Pedestrians affiliated with a single social group should be excluded from social distancing violations.
- Off-the-shelf group detectors cannot be used for our dataset, since those detectors require information such as body and head orientation, velocity, exact trajectory, etc. which cannot be obtained from our dataset, due to the high altitude of the cameras, their oblique view, and obstacles at the intersection blocking the view of the cameras.
- We designed a group detection algorithm that can detect pedestrians that belong to a single social group with the limited data that we can derive from cameras such as the ones in the COSMOS pilot site.
- Group detection module calculates the correlation between these trajectories to check if two pedestrians belong to a single social group.

Algorithm 2 Group Detection

Input: $ID_{\text{vec}}, d_{\max}, d_{\max}, \sigma_{\max}$
Output: ID_{vec} Pdestrians belong to a group
for $id \in ID_{\text{vec}}$ **do**
  $id.TimeTrj = \text{map}(id.TimeStepVec, id.Trj)$
for $(id_1, id_2) \in ID_{\text{vec}}$ **do**
  $n = 0$
  for $t = 1 : T$ **do**
    $pos_1 = id_1.TimeTrj(t),\ \ pos_2 = id_2.TimeTrj(t)$
    $d = ||pos_1 - pos_2||_2$
    if $d > d_{\max}$ **then**
      $n++$, continue
    $Corr_{\text{vec}}(id_1, id_2).\text{append}(d)$
  if $n > N_{\max}$ **then**
    continue
  $\bar{d} = \text{mean}(Corr_{\text{vec}}(id_1, id_2))$ ▷ 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 $\bar{d} < \bar{d}_{\max}$ && $\sigma < \sigma_{\max}$ **then**
    id_1 and id_2 belongs to the same group

Measurements and Evaluation

- **Dataset and Evaluation Setup**
  - We applied Auto-SDA to videos recorded from a camera deployed on the 2nd floor of the Columbia Mudd building at the COSMOS pilot site.
  - The camera is configured to record 180 sec (which is two times the signal timing cycle of the traffic lights at the intersection) videos, 5 times a day at 9 AM, 2 PM, 5:30 PM, 7:30 PM, and 10 PM.
  - Auto-SDA is deployed in one of the COSMOS edge servers. We applied it on videos recorded between June 17 and July 20, 2020 (after the lockdown), and during May 2021 (after the vaccines became broadly available).
  - We used 16 sample videos recorded before the COVID-19 outbreak (in June 2019) to evaluate the impact of the pandemic on pedestrians' density
- **Privacy Preserving**
  - The use of the videos by Columbia researchers is IRB-exempt. The recorded videos are solely used for research-related purposes, and they will not be shared in any way.



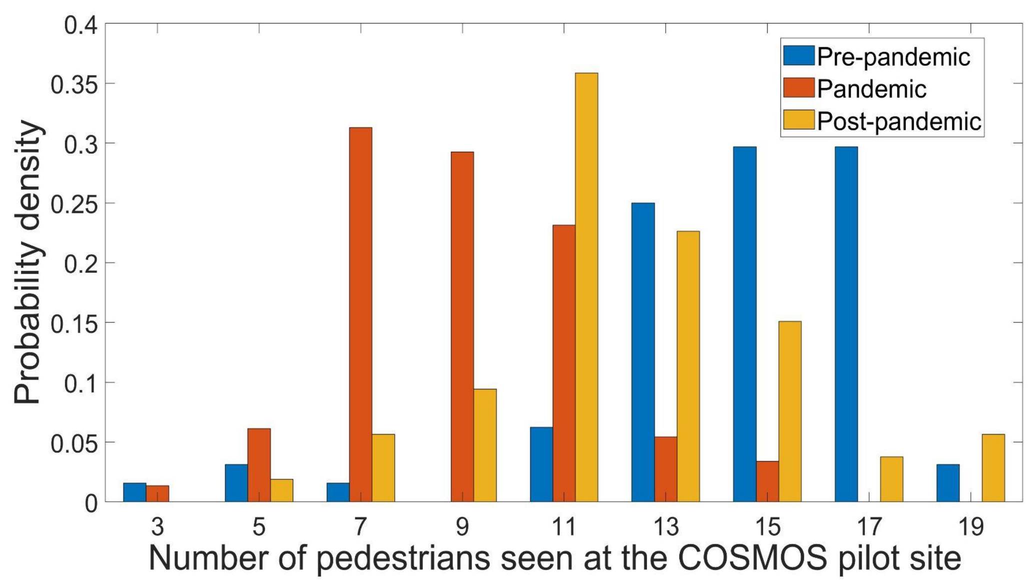
Demonstration of the performance of Auto-SDA on the sample videos collected from the COSMOS pilot site. Bounding boxes are color-coded with blue, red, and green. Blue bounding box: a social group. Red bounding box: social distancing violation. Green bounding box: distance from unaffiliated pedestrians greater than 6 ft, but less than 6.5 ft.

Comparison of Prior Work to Auto-SDA

Framework	Object Detection	Tracking	Calibration Method	On-Ground Distance Computation Error	Correction	Group Detection	Real-World COVID-19 Pandemic Impact Analysis
[1]	✓	X	Homography transformation	$\gg$ 10 cm	X	X	X
[2]	✓	✓	Pairwise L_2 norm	$\gg$ 10 cm	X	X	X
[3, 4, 5, 6]	✓	X	Planar camera persp. trans.	$\gg$ 10 cm	X	X	X
Auto-SDA	✓	✓	Multi-area calibration	< 10 cm	✓	✓	✓

Impact of the COVID-19 Pandemic on the Density of the Pedestrians

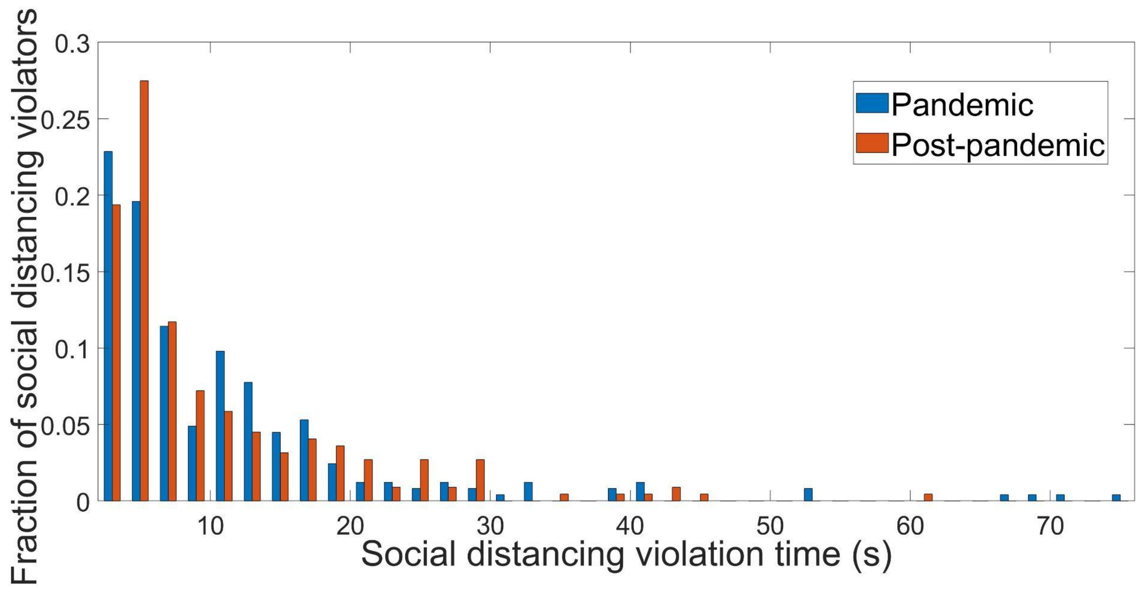
Density of the pedestrians has decreased by almost 50% after the lockdown (compared to pre-pandemic), while it has slightly increased recently.



Comparison between the density of pedestrians walking at the COSMOS pilot site in different periods.

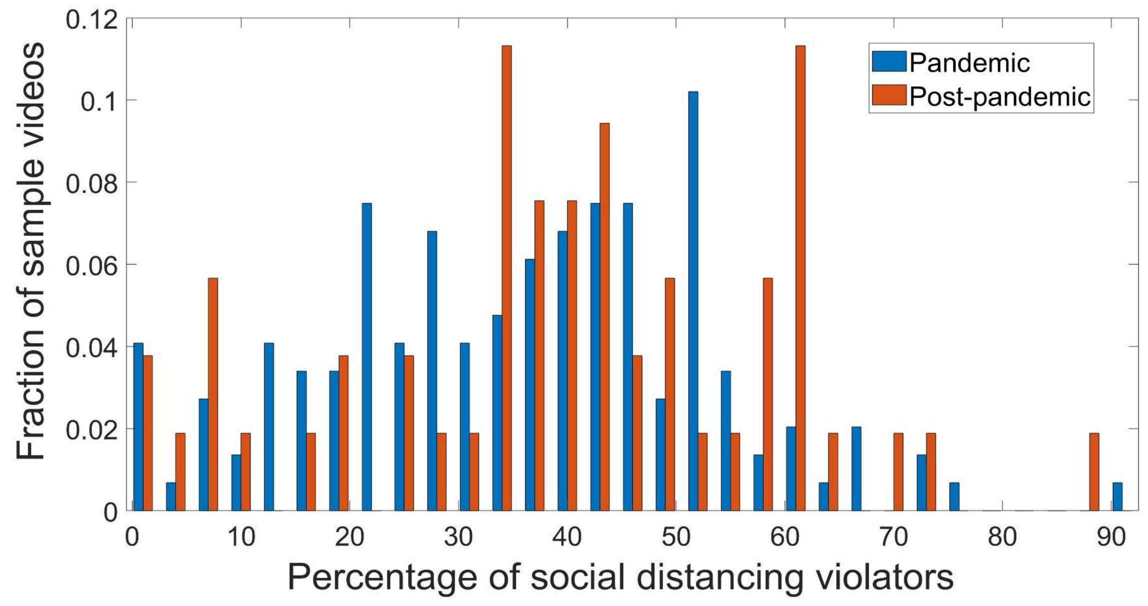
Compliance of the Pedestrians with the Social Distancing Protocols

- Social distancing violation time (mean, std):
  Pandemic, i.e., June-July, 2020: (10, 7.94) s.
  Post-pandemic, i.e., May, 2021: (10.34, 9.14) s.



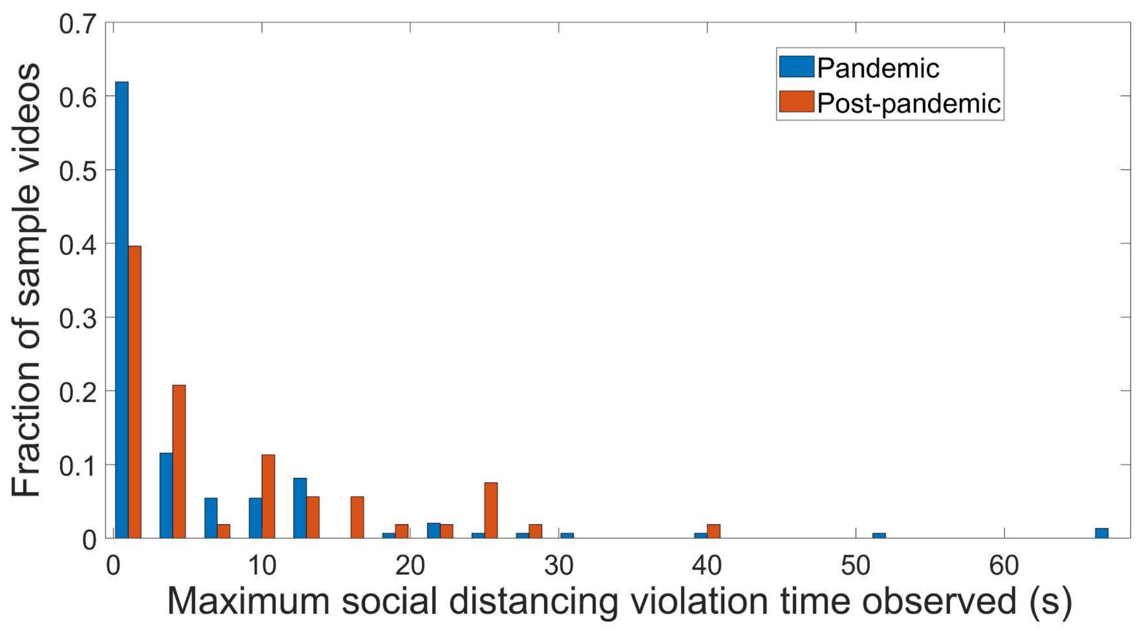
Normalized histogram of the duration of the detected social distancing violation incidents.

- Percentage of social distancing violators (mean, std):
  Pandemic, i.e., June-July, 2020: (36.2, 17.88).
  Post-pandemic, i.e., May, 2021: (40.34, 19.87).



Normalized histogram of the percentage of pedestrians considered social distancing violators in the recorded videos.

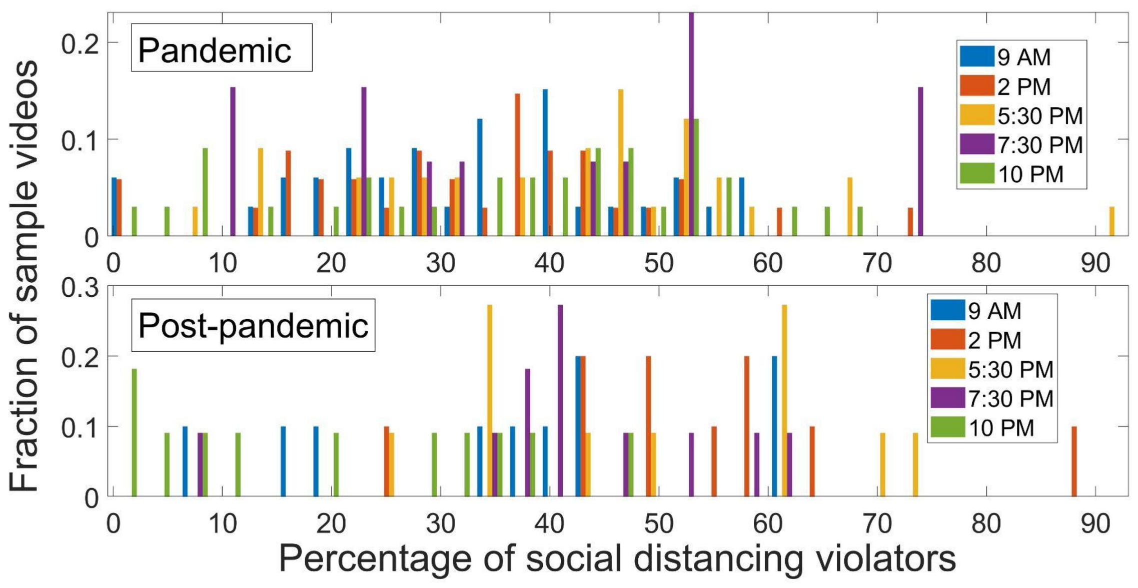
- Maximum duration of social distancing violations (mean, std):
  Pandemic, i.e., June-July, 2020: (5.99, 10.68) s.
  Post-pandemic, i.e., May, 2021: (8.5, 9.22) s.



Normalized histogram of the maximum duration of social distancing violation observed.

- Percentage of social distancing violators at different times of the day (mean, std):

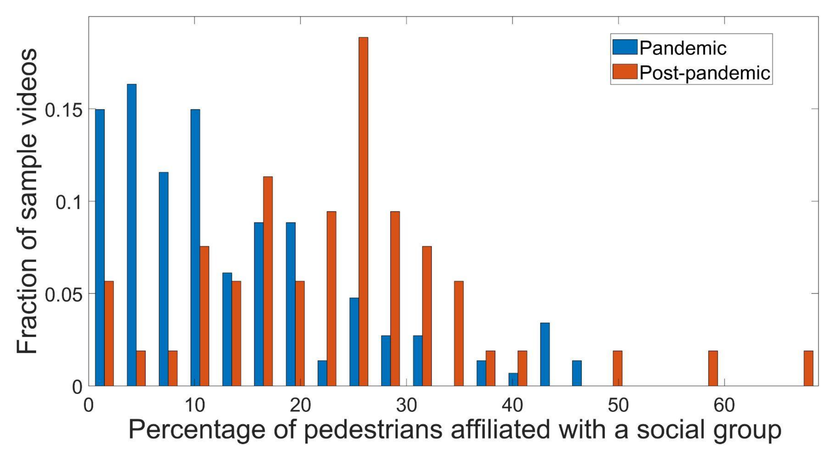
Pandemic,	Post-pandemic,
9 AM: (32.67, 15.39).	9 AM: (36.69, 18.46).
2 PM: (33.3, 16.01).	2 PM: (53.62, 16.37).
5:30 PM: (41.39, 18.28).	5:30 PM: (50.36, 17.02).
7:30 PM: (40.27, 21.43).	7:30 PM: (41.63,14.34).
10 PM: (37.02, 18.82).	10 PM: (20.29, 16.3).



Normalized histogram of the number of pedestrians violating social distancing protocols in different times of the day.

Impact of the COVID-19 Pandemic on the Pedestrians' Social Behavior

- Percentage of pedestrians affiliated with a social group (mean, std):
  Pandemic, i.e., June-July, 2020: (13.47, 11.42).
  Post-pandemic, i.e., May, 2021: (22.03, 10.15).



Normalized histogram of the percentage of pedestrians affiliated with a social group.

Future Work

- Design and implementation of privacy-preserving methods.
- Integration of information from multiple cameras and sensors.
- Design of real-time algorithms, and extensive evaluation as the social distancing policies change.
- Distributed edge/cloud video analytics.
- Support cloud-connected vehicles to overcome the limitations of autonomous vehicles.
- Enable vehicles to wirelessly share in-vehicle sensor data with other vehicles and the edge/cloud servers.
- Low-latency AR edge/cloud.
- Managing heterogeneous bandwidth and computing resources, and providing seamless service mobility.
- Provide performance guidelines for future edge/cloud deployments.

Acknowledgements

I thank Ivan Seskar, Jakub Kolodziejski, and Michael Sherman (Rutgers/WINLAB) and Professors Javad Ghaderi, Zoran Kostic, and Gil Zussman (Columbia). This work was supported in part by NSF grants CNS-1827923, OAC-2029295, NSF-BSF grant CNS-1910757, NSF grant CNS-2038984, and ARO grant W911NF1910.

References

1. Mahshid Ghasemi, Zoran Kostic, Javad Ghaderi, and Gil Zussman. "Auto-SDA: Automated video-based social distancing analyzer." In Proc. ACM HotEdgeVideo'21..
2. Dongfang Yang, Ekim Yurtsever, Vishnu Renganathan, Keith A Redmill, and Ümit Özgüner. 2020. A vision-based social distancing and critical density detection system for COVID-19. arXiv preprint arXiv:2007.03578 (2020), 24–25.
3. Narinder Singh Punn, Sanjay Kumar Sonbhadra, and Sonali Agarwal. 2020. Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLOv3 and DeepSort techniques. arXiv preprint arXiv:2005.01385 (2020).
4. John Betancourt. 2020. Social Distancing Analyser. https://github.com/JohnBetaCode/Social-Distancing-Analyser
5. Deepak Birla. 2020. Social Distancing AI. https://github.com/deepak112/Social-Distancing-AI.
6. Marco Cristani, Alessio Del Bue, Vittorio Murino, Francesco Setti, and Alessandro Vinciarelli. 2020. The visual social distancing problem. arXiv preprint arXiv:2005.04813 (2020).
7. Tom Farrand. 2020. Social Distancing. https://github.com/FarrandTom/social-distancing.
8. Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020).
9. Sebastiano Vascon and Loris Bazzani. 2017. Group detection and tracking using sociological features. In Group and crowd behavior for computer vision. Elsevier, 29–66.
10. Dipankar Raychaudhuri, Ivan Seskar, Gil Zussman, Thanasis Korakis, Dan Kilper, Tingjun Chen, Jakub Kolodziejski, Michael Sherman, Zoran Kostic, Xiaoxiong Gu, et al. 2020. Challenge: COSMOS: A city-scale programmable testbed for experimentation with advanced wireless. In Proc. ACM MobiCom'20.