Auto-SDA: Automated Video-based Social Distancing Analyzer

Mahshid Ghasemi, Zoran Kostic, Javad Ghaderi, Gil Zussman

Electrical Engineering, Columbia University

{mahshid.Ghasemi,zk2172,jghaderi,gil.zussman}@columbia.edu

COSMOS

wim.net
Wireless & Mobile Networking Lab

NEAL
Network Algorithms

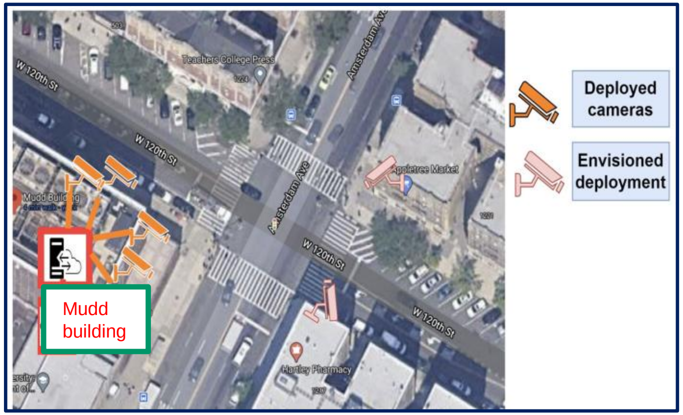
Columbia University in the City of New York

COVID-19 and Social Distancing

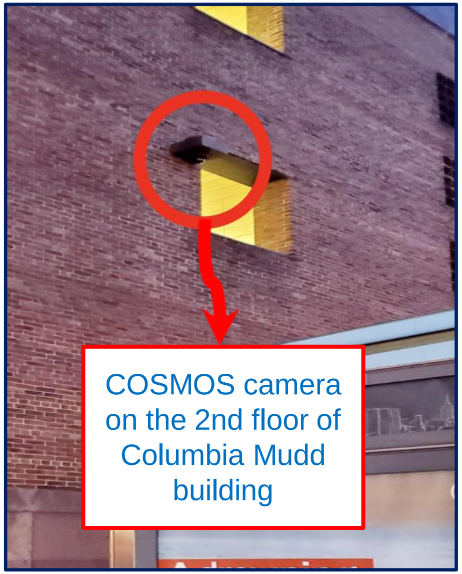
Social Distancing Analysis Using the COSMOS Testbed

Automated video-based **S**ocial **D**istancing **A**nalyzer (Auto-SDA)

- Measures compliance with social distancing policies.
- Evaluated using the **COSMOS testbed** deployed in West Harlem, NYC.
- Used a camera deployed on the **2nd floor of Columbia's Mudd building** looking at the **COSMOS site**[1]**.**



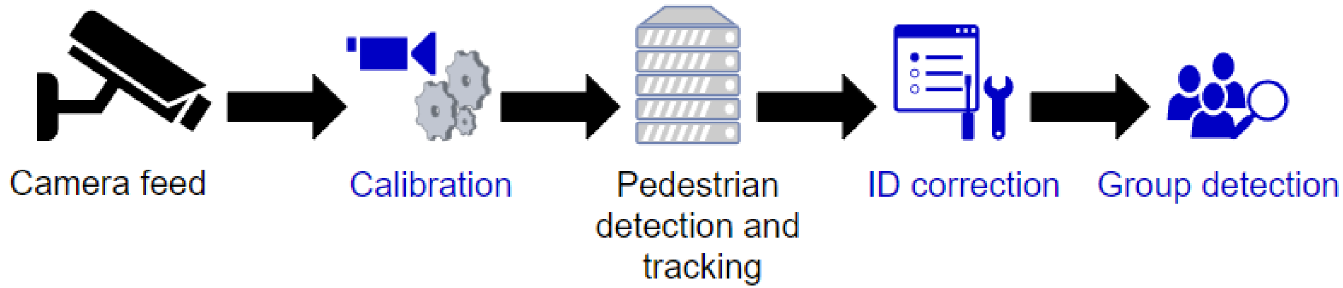
The NSF PAWR COSMOS site at 120th St. And Amsterdam Ave. intersection, NYC.



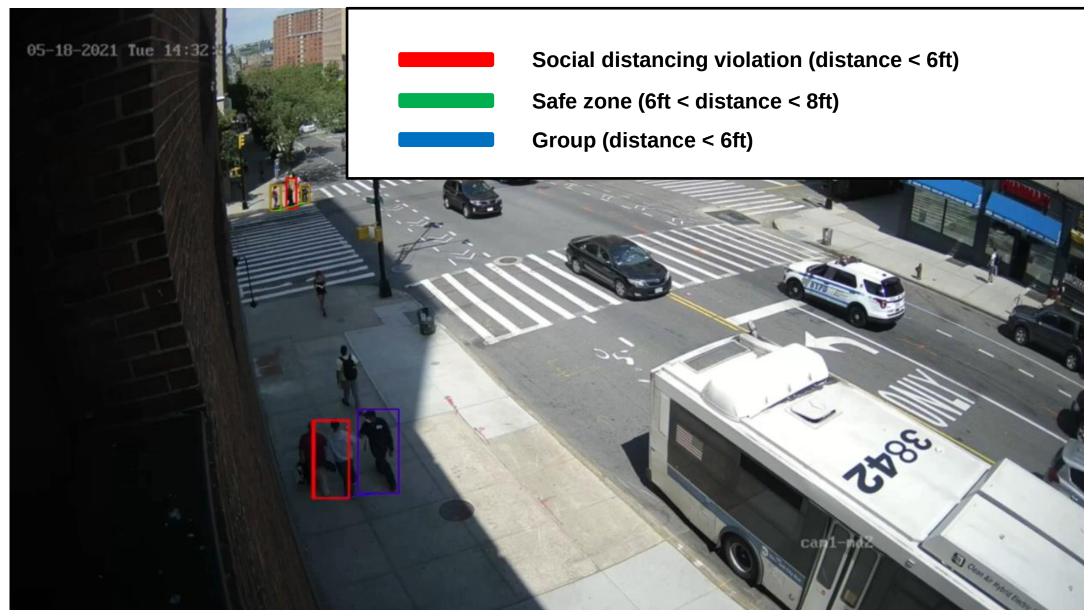
Viewpoint of the camera deployed on the 2nd floor of the Columbia's Mudd building.

[1] https://wiki.cosmos-lab.org/wiki/Hardware/Cameras

Automated Social Distancing Analyzer (Auto-SDA)



- **Calibration:** Converts 2D on-image distances to 3D on-ground distances.
- **Object detection and tracking:** Locates the pedestrians and assigns an ID to each of them.
- **ID correction:** Removes the redundant IDs generated by the tracker and extract the trajectory of each pedestrian.
- **Group detection:** Excludes the pedestrians affiliated with a single social group from social distancing violations.



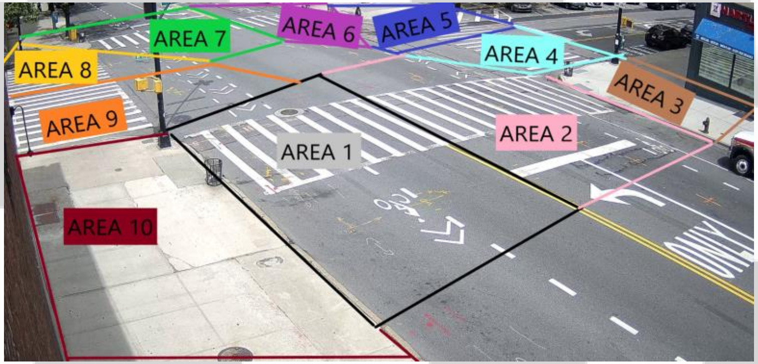
Multi-area Camera Calibration: Intrinsic Parameters

- Camera calibration is **required** to convert 2D on-image distances to 3D on-ground distances.
- Camera calibration means calculating the **intrinsic** and **extrinsic** parameters of the camera.
- Intrinsic parameters can be obtained using a **checkerboard.**
- Intrinsic parameters **does not** depend on view point of the camera.
- More than 20 images of the checkerboard in different poses were provided to the OpenCV library to obtain the intrinsic parameters of the camera.

Calibration of the COSMOS cameras using a checkerboard:

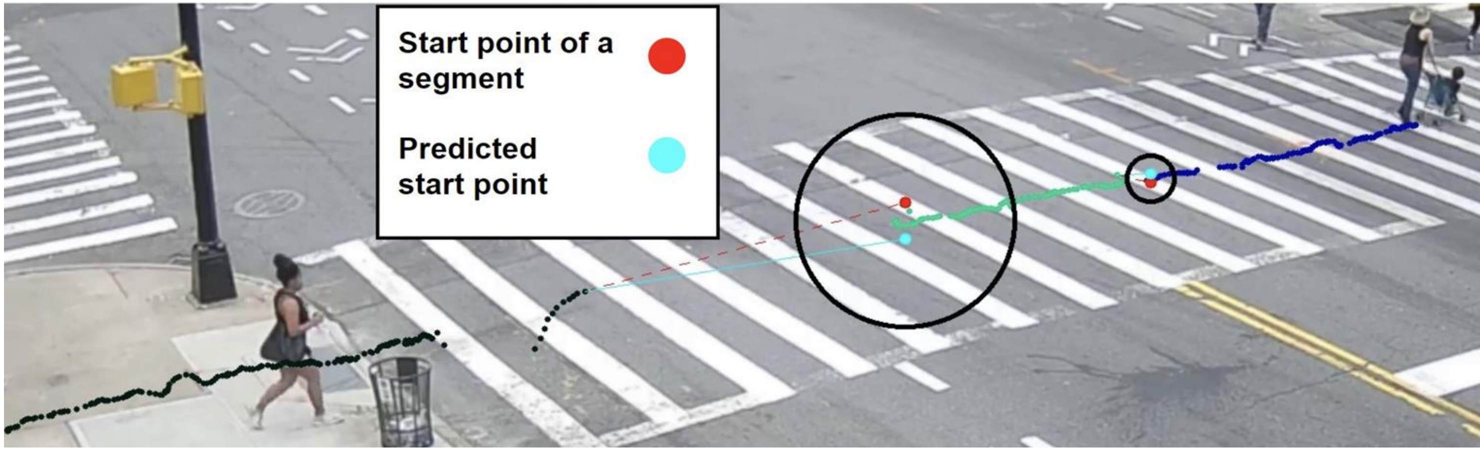
Multi-area Camera Calibration: Extrinsic Parameters

- Standard camera calibration methods lead to **inaccurate on-ground distance** computation.
- Calibration module **splits the view** of the camera into **multiple areas**.
- The **extrinsic parameters** for each area are computed individually.
- **Multi-area camera calibration** can obtain on-ground distances with **less than 10 cm error**.



	Calculated Distance from Different Methods (cm)			
Pixel Coordinates of a Pair of Points on a 4K Frame	Ground-truth	Multi-area Calibration	Homography Transformation	Planar Camera Perspective Transformation
[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

ID Correction



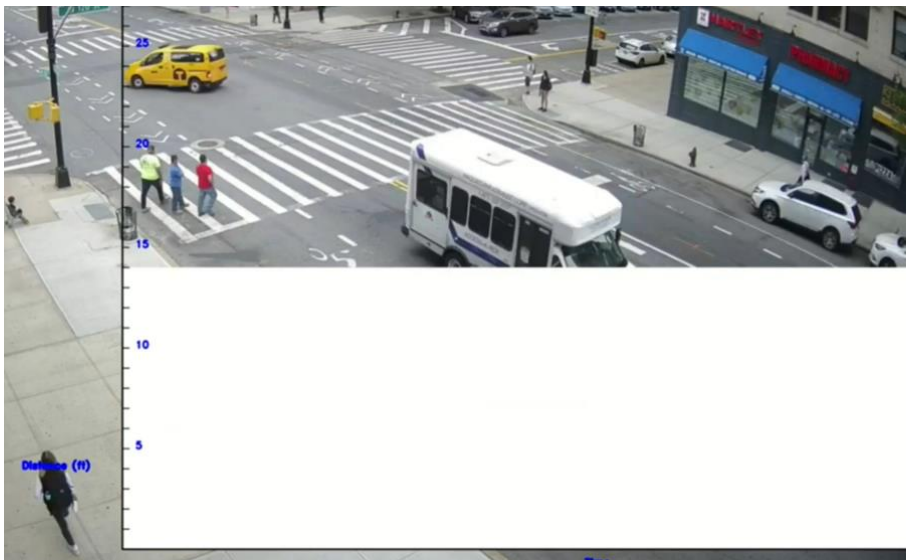
- **High altitude** and **oblique view** of traffic cameras (including the COSMOS camera), and **obstacles** degrade the performance of the tracker.
- Compensates for the inaccuracies of the tracker.
- **Removes the redundant IDs** and extracts the **entire trajectory** of the pedestrians.
- Uses **Linear Regression (LR)** to find trajectory segments.

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**
  $t1 \leftarrow id_1.TimeStamp.StopTime$
  $t2 \leftarrow id_2.TimeStamp.StartTime$
  $p_1 \leftarrow id_1.TailEst.\text{at}(t = t_2), \quad p_2 \leftarrow id_1.Trj.\text{at}(t_2)$
  $v_1 \leftarrow id_1.TailDir, \quad v_2 \leftarrow id_2.HeadDir$
  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

Group Detection

- Excludes social groups from social distancing violation.
- **Off-the-shelf group detectors** require detailed information such as body and head orientation, velocity, exact trajectory, etc.
- In realistic deployments, such as COSMOS cameras, **details cannot be obtained**.
- **Detects** the social groups **with limited data** from the cameras.



Demonstration of the group detection algorithm on a recorded video from the camera on the 2nd floor of Columbia Mudd building.

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

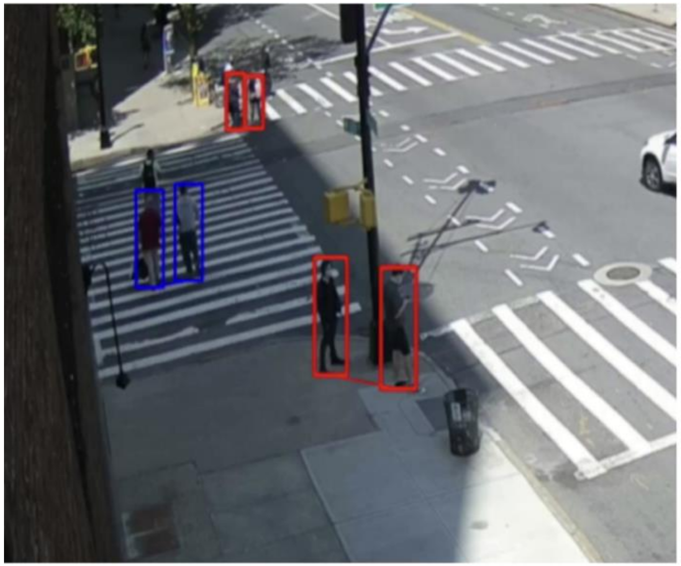
Comparison of Prior Work to Auto-SDA

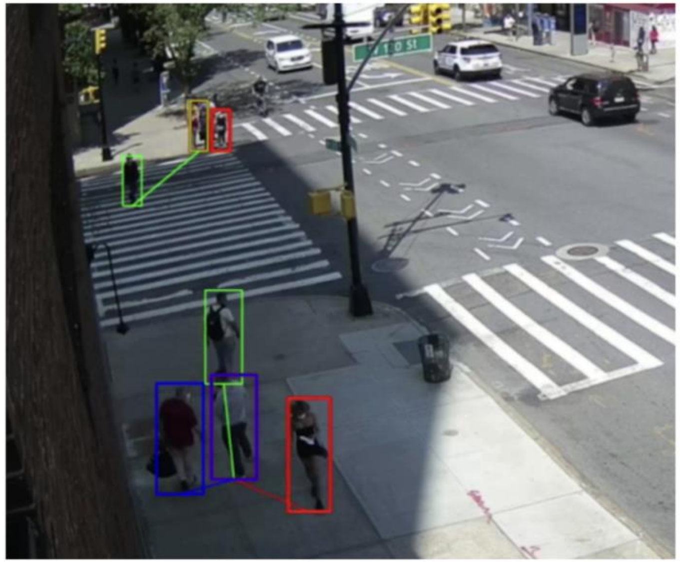
Framework	Object Detection	Tracking	Calibration Method	On-Ground Distance Computation Error	Correction	Group Detection	Real-World COVID-19 Pandemic Impact Analysis
[1, 2]	✓	X	Homography transformation	$\gg$ 10 cm	X	X	X
[3]	✓	✓	Pairwise L_2 norm	$\gg$ 10 cm	X	X	X
[4]	✓	✓	Planar camera persp. trans.	$\gg$ 10 cm	X	X	X
[4, 5, 6, 7]	✓	X	Planar camera persp. trans.	$\gg$ 10 cm	X	X	X
Auto-SDA	✓	✓	Multi-area calibration	$<$ 10 cm	✓	✓	✓

1. Dongfang Yang, Ekim Yurtsever, Vishnu Renganathan, Keith A Redmill, and Ümit Özgüner. 2021. A vision-based social distancing and critical density detection system for COVID-19. Sensors (2021).
2. Sergio Saponara, Abdussalam Elhanashi, and Alessio Gagliardi. Implementing a real-time, AI-based, people detection and social distancing measuring system for Covid-19. JRIP (2021)
3. Narinder Singh Punn, Sanjay Kumar Sonbhadra, and Sonali Agarwal. Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLOv3 and DeepSort techniques. arXiv preprint arXiv:2005.01385 (2020).
4. Anupriya Koneru, P Ragini, M Sri Vastav, and K Jashnavi. A Real-Time Solution for Social Distance Detection in COVID-19 Pandemic. ICICCS (2021).
5. John Betancourt. Social Distancing Analyser. https://github.com/JohnBetaCode/Social-Distancing-Analyser (2020).
6. Deepak Birla. Social Distancing AI. https://github.com/deepak112/Social-Distancing-AI (2020).
7. Marco Cristani, Alessio Del Bue, Vittorio Murino, Francesco Setti, and Alessandro Vinciarelli.The visual social distancing problem. IEEE Access (2020).
8. Tom Farrand. Social Distancing. https://github.com/FarrandTom/social-distancing (2020).

Measurements and Evaluation

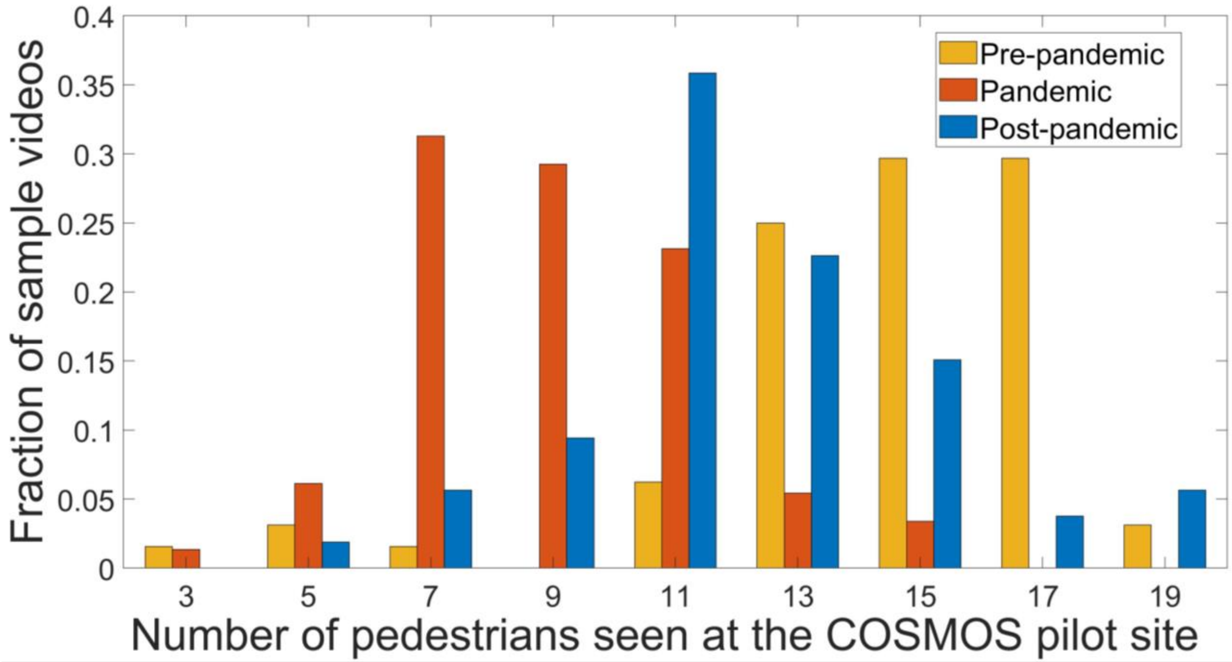
- Evaluating the impact of COVID-19 on pedestrians behavior using recorded videos from the COSMOS camera.
- Recorded videos consist of:
  - **Pre-pandemic** videos that were opportunistically recorded before the pandemic in **June 2019**.
  - **Pandemic** videos recorded between **June 17 to July 20, 2020 (soon after the lockdown)**.
  - **Post-vaccine** videos recorded in **May 2021 (after the vaccine became broadly available)**.





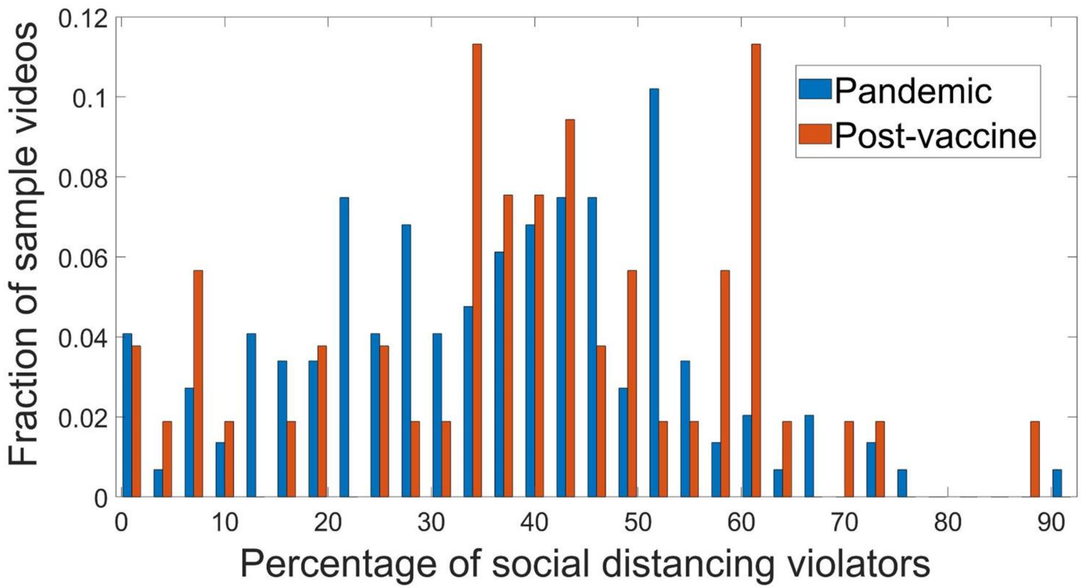
Impact of the Pandemic on Pedestrians' Density

- Density of the pedestrians has **decreased after the lockdown** (compared to pre-pandemic).
- After the availability of the **vaccine** the density has **slightly increased**.
  - **Pre-pandemic,** i.e., June, 2019
  - **Pandemic**, i.e., June-July, 2020
  - **Post-vaccine**, i.e., May, 2021



Compliance with Social Distancing Protocols

- Social distancing violation: when a pedestrian maintains less than 6 ft distance from other pedestrians with whom he/she is not walking.
- Percentage of social distancing violators (average, std):
  - **Pandemic**, i.e., June-July, 2020: (36.2, 17.88).
  - **Post-vaccine**, i.e., May, 2021: (40.34, 19.87).

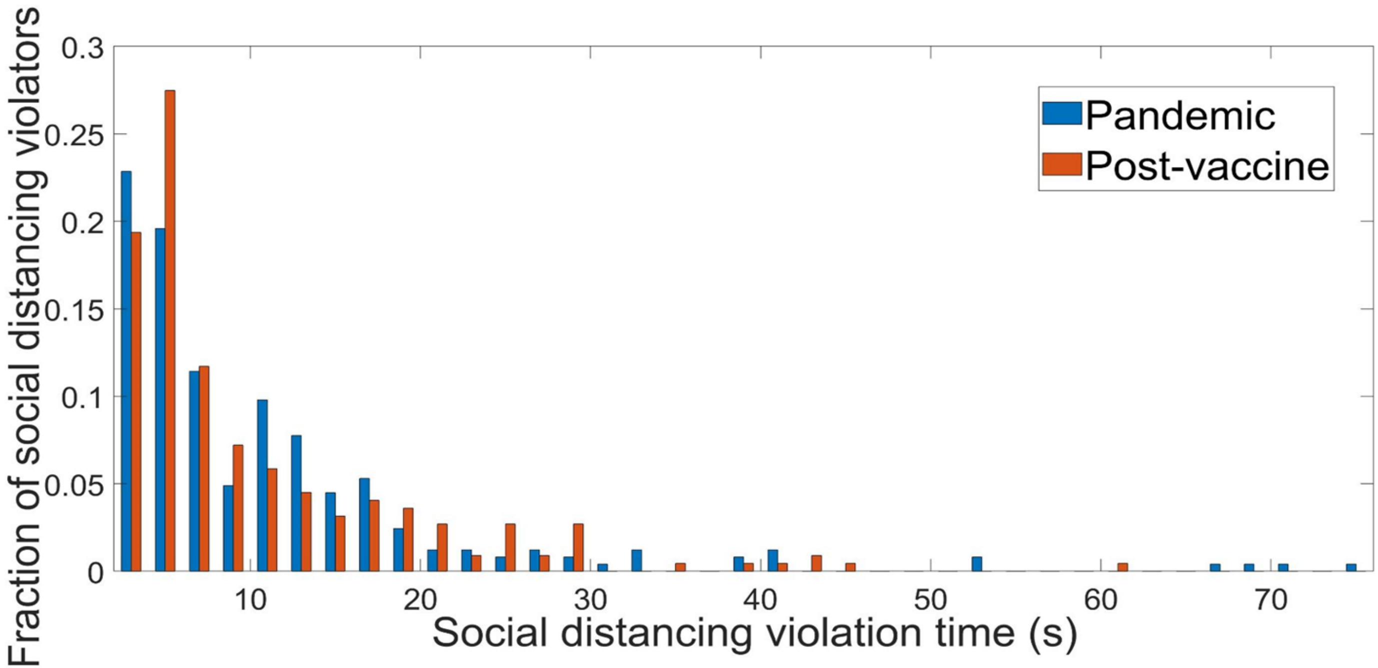


Normalized histogram of the percentage of pedestrians considered social distancing violators in the recorded videos.

Compliance with Social Distancing Protocols

Measurements of social distancing violation time (average, std):

- **Pandemic**, i.e., June-July, 2020: (10, 7.94) s.
- **Post-vaccine**, i.e., May, 2021: (10.34, 9.14) s.

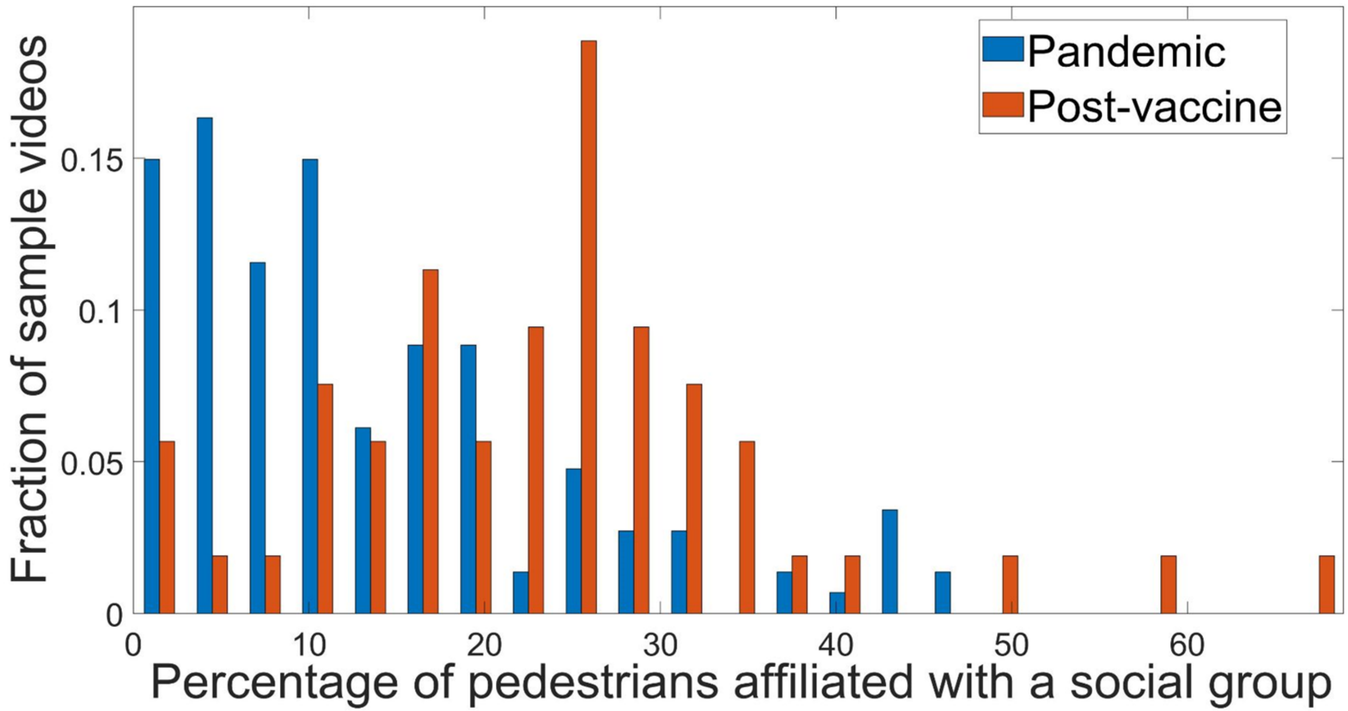


Normalized histogram of duration of the social distancing violations incidents.

Compliance with Social Distancing Protocols

Percentage of pedestrians affiliated with a social group (average, std):

- **Pandemic**, i.e., June-July, 2020: (13.47, 11.42).
- **Post-vaccine**, i.e., May, 2021: (22.03, 10.15).



Normalized histogram of duration of percentage of pedestrians in a social group.

Conclusion and Future Work

Auto-SDA modules enable a generic object detection and tracker model to be used as a **social distancing analyzer system**:

- **Multi-area calibration** computes the on-ground distances between pedestrians with **high accuracy**.
- **ID correction** rectifies the output of the tracker model.
- **Group detection** excludes social groups from social distancing violation.
- Used **real-word videos** to evaluate the system and measure the impact of social distancing policies on pedestrian's social behavior.

Future work:

- Design and implementation of **privacy-preserving** methods.
    - Extension of Auto-SDA (B-SDA) will be presented in the demo session which uses **bird's eye view camera**.
- Integration of information from **multiple cameras** and sensors.
- Design of real-time algorithms, and extensive evaluation as the social distancing policies change.