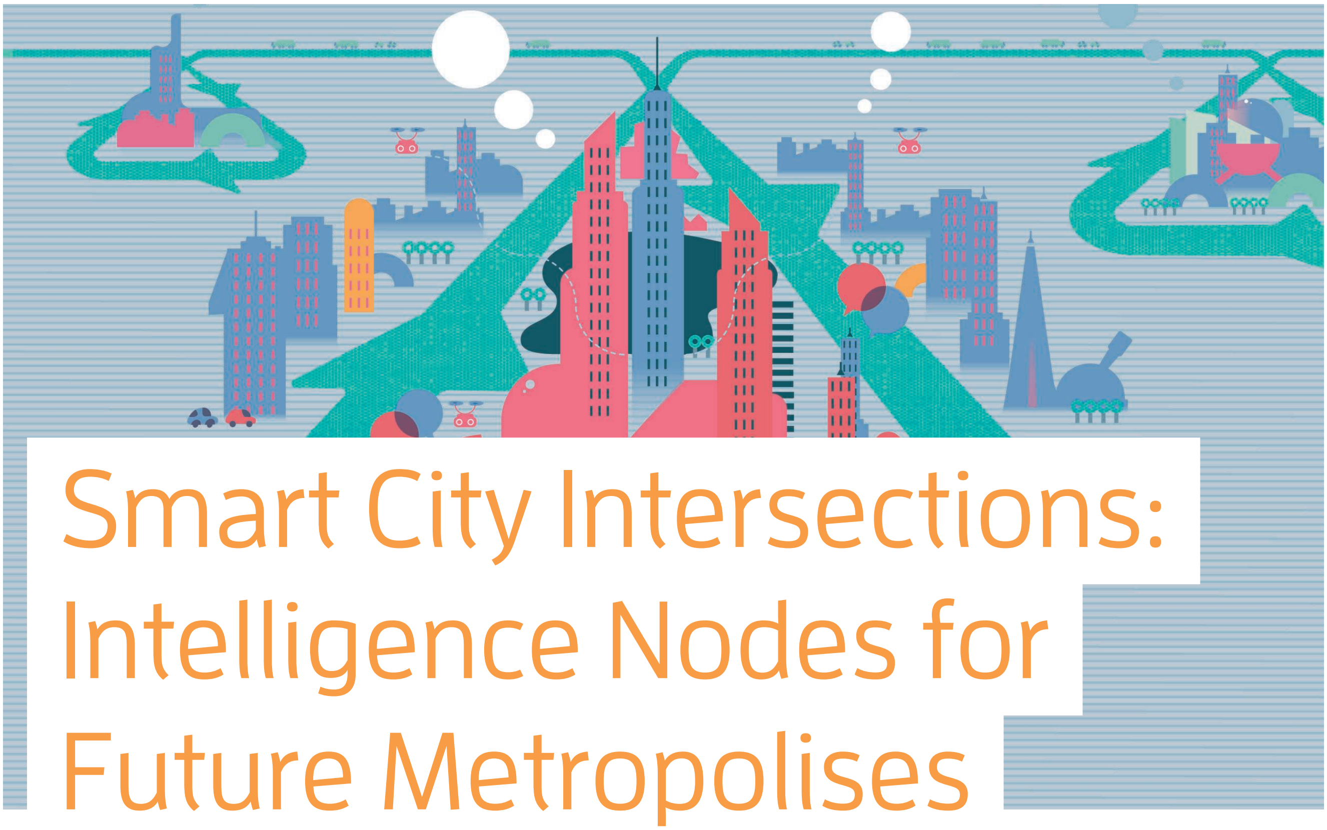

Smart City Intersections: Intelligence Nodes for Future Metropolises

Zoran Kostić, Columbia University
Alex Angus, Qualcomm, Inc.
Zhengye Yang and Zhuoxu Duan, Rensselaer Polytechnic Institute
Ivan Seskar, Rutgers University
Gil Zussman, Columbia University
Dipankar Raychaudhuri, Rutgers University

This article explores city intersections as intelligence nodes using high-bandwidth, low-latency services for providing privacy-preserving smart city applications. COSMOS testbed experiments using edge-computing-based artificial-intelligence techniques are reported, for monitoring of pedestrians, cloud-connected vehicles, and traffic management.

Smart cities should be built with the primary goal of providing social good as defined by local communities.[1,2] Contemporary technologies provide a plethora of components to support human-centered design of future metropolises. Privacy, security, and local data governance on one hand, and optimization of bandwidth, computational resources, and latency on the other hand, implicate traffic intersections as excellent locations for smart city intelligence nodes.

Traffic intersections can support smart city features and traffic dynamics by utilizing available power supply

Digital Object Identifier 10.1109/MC.2022.3206273
Date of current version: 15 November 2022

and communications infrastructure to enable interconnections and computational collaboration among neighboring intelligence nodes. The nodes will be equipped with artificial intelligence (AI)-enabled edge-computing[3] and communications equipment to facilitate automated low-latency data harvesting, inference, and decision making. This will enable the development of technologies like cloud connected vehicles, vehicle to infrastructure communications, and advanced sensor-based tools for alerting pedestrians and assisting handicapped individuals. Future applications will require intense AI-enabled computation, very high communication bandwidths, and ultralow latencies.

We report the results of research on low-latency real-time applications for smart city intersections in metropolises and architectures, components, and methods for building intelligent intersection nodes. The research utilizes COSMOS, an experimental testbed located in New York City.[4]

SMART CITY INTERSECTIONS

The focus of this article is low-latency high-bandwidth applications for smart city intersections. We explore technological components needed to support privacy-preserving real-time applications, such as collaborative control of cloud-connected vehicles and active pedestrian alert and assistance, where the primary sensors are multiple high-resolution surveillance cameras. One of the key tasks for video-based applications is to detect and track objects in an intersection with high accuracy. We explore methods to achieve real-time processing in smart city intersection applications defined by end-to-end latencies under 33.3 ms. This includes 1) sensor data acquisition, 2) communication among end-users, sensors, and edge cloud, 3) AI-based inference computation, and 4) providing feedback to participants in the intersection. The envisioned "radar-screen" application is intended to broadcast the positions and velocities of objects to intersection participants in real time.

Privacy

Smart-city implementations prior to year 2022 indicate that privacy and data security are the key concerns impeding successful large-scale deployments. Privacy concerns are amplified when video recordings are a part of data acquisition and processing. The COSMOS research program has a strong community outreach component, exemplified by multiyear activities on running National Science Foundation (NSF) Research Experience and Mentoring (REM) and Research Experiences for Teachers (RET) programs where teachers from Harlem and other New York City schools get training and participate in developing science, technology, engineering, and mathematics educational material for students in underprivileged schools (https://www.cosmos-lab.org/outreach/[5]). Our guiding approach to privacy is to integrate local communities into the data governance process. We plan to develop technologies that enable the communities to define and control data acquisition and processing supported by edge computing and temporary data storage paradigms. This article describes some technological components which would need to be managed in collaboration with local communities, such as blurring of faces and license plates.

> IF THE VEHICLE'S BRAKES COULD BE ACTIVATED WITHIN THAT TIME, IT IS CONCEIVABLE THAT LIFE-THREATENING TRAFFIC ACCIDENTS WOULD BE AVOIDED.

Real-time interactions

An important goal of smart city deployments is to improve the safety of pedestrians and other participants. Even in the most congested cities it is desirable to replace human drivers with safer self-driven vehicles. This motivates the concept of cloud-connected vehicles that interact with city infrastructure to improve their ability to navigate and requires exceptionally low closed-loop latencies associated with security-critical real-time actions. This article explores latencies which are inherent in camera-based sensor data acquisition and processing.

Real-time for safety-critical applications. Extracting intelligence that indicates a potential collision and providing feedback to vehicles or pedestrians presents computational and latency challenges. City street dynamics are determined by vehicles traveling at velocities between 0 and 100 km/h. If we consider for example a vehicle traveling at 10 km/h, an arguably reasonable speed within congested intersections, the vehicle is moving at approximately 3 m/s. If we divide 3 m/s by the standard frame rate of a conventional video equal to 30 fps, the result is a vehicle movement of 10 cm, the distance traveled in 33.3 ms. If the vehicle's brakes could be activated within that time, it is conceivable that life-threatening traffic accidents would be avoided. This approximate calculation motivated us to investigate what is needed to support latencies below 33 ms—this is an aggressive target, which is not achievable by contemporary chain of sensing, video coding, communications, and computing technologies.

CONVENTIONAL VIDEO STREAMING PROTOCOLS MAY BE INADEQUATE FOR ACCOMPLISHING VERY LOW LATENCIES, SO RESEARCH INTO EDGE-STREAMING PROTOCOLS IS AN APPEALING TOPIC.

Sensor latencies. Smart city sensors will have a wide range of operational frequencies and data acquisition bandwidths. CO_2 sensors may collect several bytes per hour, whereas high-resolution cameras may stream data in compressed form at tens of megabits per second, or in uncompressed form at several gigabytes per second. Low-cost CMOS imaging sensors have latencies of several milliseconds, which are low enough not to obstruct the closed-loop target of 1/30 s. Internet Protocol (IP) cameras use video encoding and streaming protocols that, because of interframe coding, may have buffers requiring hundreds of milliseconds to decode; this process severely impedes the ability to provide closed-loop services faster than 33.3 ms.

Communications latencies. Communications and networking latencies are determined as much by speed of physical media as they are driven by protocols at the application layer. The COSMOS optical network can provide up to 100 Gb/s, offering almost unlimited raw speed. On the other hand, conventional streaming of high-resolution videos can create hundreds of milliseconds of latency. This suggests that video processing and inference are best done at the "extreme" edge—right next to the video sensor. More interestingly, this motivates research on integrated coding and video transmission protocols optimized for ultralow latency transmission of videos over high bandwidth edge communications infrastructure.

Inference and decision latencies. Inference latencies come from video preprocessing and deep learning (DL) algorithms for object detection and multiple-object tracking (MOT). The training of DL models is done offline and does not impact latencies for real-time interactions. Both published work and our own studies indicate that contemporary GPUs within specialized pipelines such as NVIDIA TensorRT and DeepStream can deliver speeds above 30 fps for object detection and tracking. We previously showed that inference speed varies as a function of input resolution and actual device capabilities, but we assess that inference computation will not be a bottleneck in meeting our real-time latency target.

The decision process is defined as a higher level of intelligence built on top of object detection and tracking. For example, this process would deduce the implications of a pedestrian being on a trajectory to intersect with a speedy vehicle and create a warning (or even a command) for the pedestrian or vehicle. Computational needs for this type of processes are subject to ongoing studies, but it is expected that the corresponding latency would be less than a millisecond.

COSMOS experimental testbed

New York City is an example of a busy metropolis which provides formidable challenges for the deployment of smart city technologies. Busy urban traffic intersections have a large number of vehicles and pedestrians moving in many directions at various speeds, often with chaotic or unpredictable behavior. Furthermore, obstructions like building corners, parked vehicles, and construction equipment present difficulty to autonomous vehicle sensors requiring further advancements in traffic intersection-based automation of monitoring, measuring, learning, and feedback.

The COSMOS testbed, NSF-funded Cloud Enhanced Open Software Defined Mobile Wireless Testbed for City-Scale Deployment,[4] provides an experimentation platform for applications and architectures to support intelligence nodes of future metropolises. For our research, we use the COSMOS pilot site located at Columbia University, in New York City, at the intersection of the 120th Street and Amsterdam Avenue. The pilot node includes two street-level and two bird's eye cameras, as illustrated in Figure 1. The COSMOS edge cloud servers can run real-time algorithms for detection and tracking of objects in the intersection to monitor and manage traffic flow and pedestrian safety. The node is equipped with an optical x-haul transport system that connects AI-enabled edge computing clusters. This allows for baseband processing with massively scalable CPU and GPU resources with field-programmable gate array assist, which can also support software-defined radios. Four technology layers are provided for experimentation: the user device layer, radio hardware and front-haul network resources, radio cloud, and general-purpose cloud.

BUILDING BLOCKS OF INTELLIGENT NODES

As of 2022, individual technological modules for implementing the vision of smart cities exist in the form of low-power chips, high-bandwidth modems, wired and wireless networks, and GPUs for machine learning (ML) and DL. However, major challenges exist in the domains of privacy preservation, security, intelligent decision making, system integration, and in the interactions between technology and social good.

Sensors

Sensors range from dozens of low-rate Internet of Things (IoT)-based devices collecting data about pollution to several high-resolution lidars and cameras providing real-time feeds. Multimodal data aggregation and collaborative intelligence are research topics of notable importance to smart intersection nodes.[6]

Networking

For high-bandwidth applications, networking at one intersection has to support wireless and wired connectivity from half a dozen infrastructure-installed cameras. Whereas coded video from a conventional IP-camera may require subhundred Mb/s, experimentation with ultra-low latency provides motivation to send raw video at several Gb/s per camera. Support for cloud-connected vehicles could require harvesting videos and other data from each vehicle wirelessly, in either raw or meta format. Conventional video streaming protocols may be inadequate for accomplishing very low latencies, so research into edge-streaming protocols is an appealing topic.

Edge computing

Smart city intersection applications require substantial computational resources, demand minimal latencies, and their functionality can be constrained to a limited geographical area. Furthermore, data privacy, security, and local data governance are of utmost importance. This strongly implicates edge computing as the right modality. Two forms of edge computing can be used. In the extreme, AI-based computing can be done on devices located at the sensors, such as Nvidia Jetson Nanos or ML-enabled ARM M1-M4 processors integrated into IoT chips. On the other hand, a more powerful computing node can be located in a facilities room of a building at the intersection. The node is then connected to sensors by high-speed wireless, wired, or optical infrastructure. To support low latencies from sensors to actuators via AI computing, an edge computing node has to be integrated tightly with the network communications infrastructure.

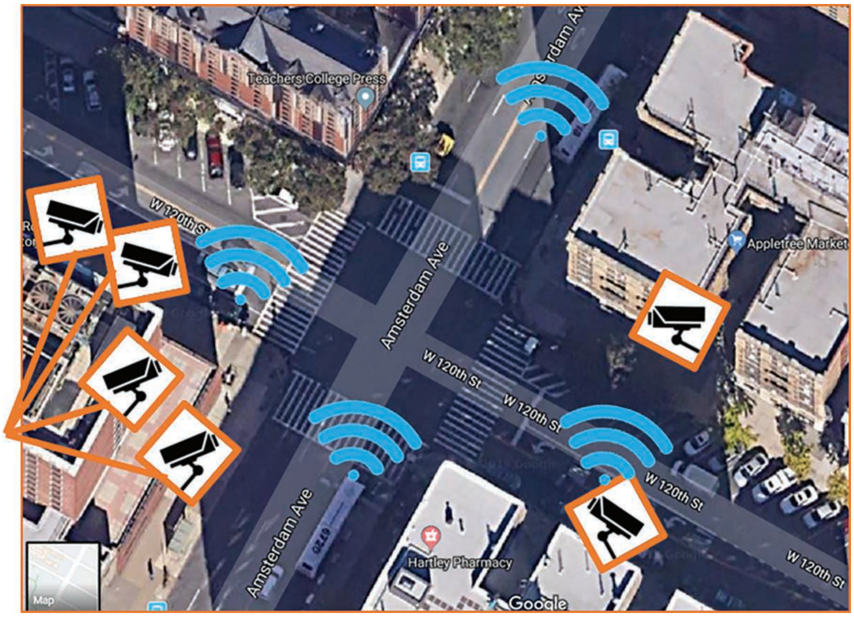

FIGURE 1. The COSMOS pilot site with cameras and edge-cloud nodes.

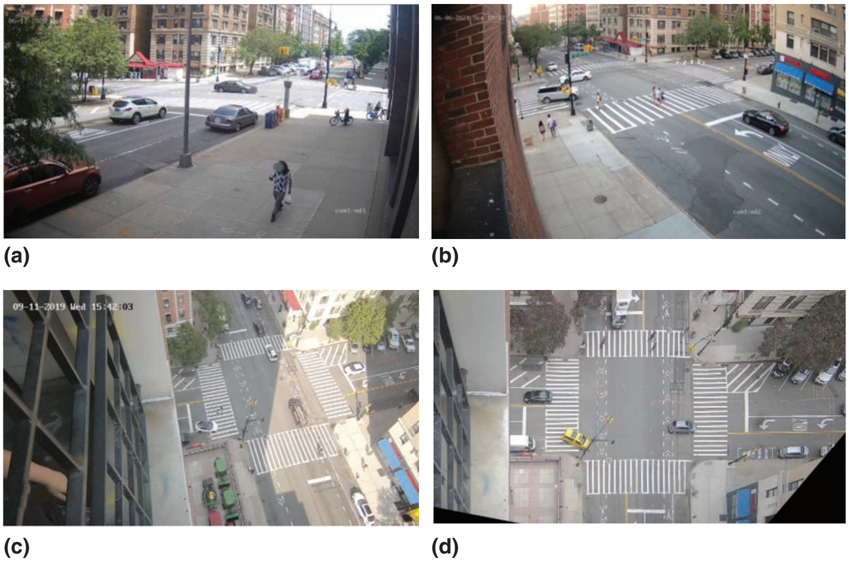

FIGURE 2. The COSMOS testbed camera views. (a) First-floor camera, 120th St. (b) Second-floor camera, Amsterdam Ave. (c) 12th-floor camera, Amsterdam Ave. (d) Calibrated 12th-floor camera.

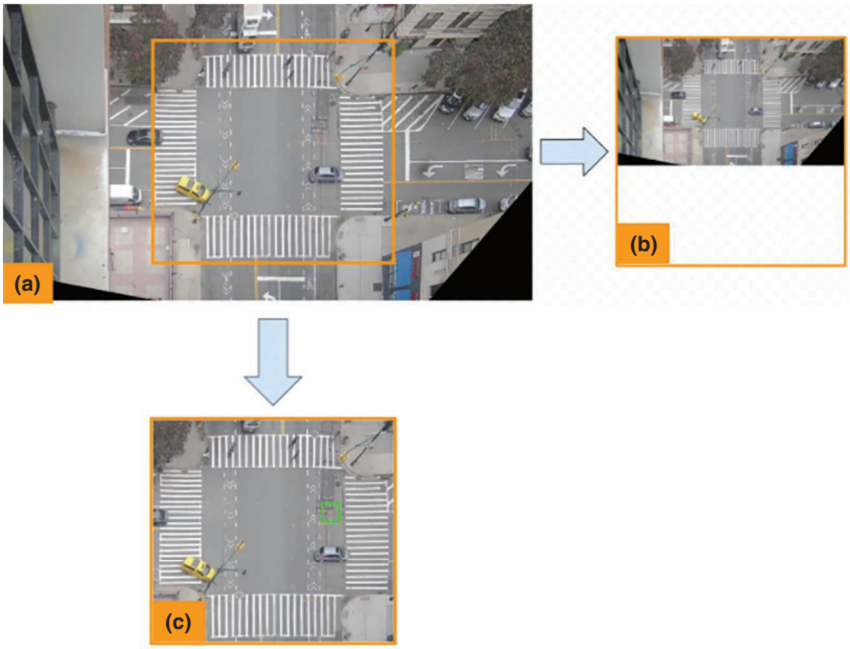

FIGURE 3. (a) Calibrated 16:9 native frame. (b) 16:9 frame squared using zero-padding. (c) Square cropped frame.

AI-enabled data processing

Intelligent tasks supporting smart city intersections are varied in complexity: CO_2 sensors generate several bytes once per hour, whereas high-resolution cameras in our studies generate megabits per second to be analyzed by visual DL models for object detection, tracking, and intelligent decisions for actuators. Automation and AI are crucial to scale systems for highly congested traffic intersections. Off-the-shelf AI models must be modified and retrained to accommodate the peculiarities of smart city intersection applications—one example being the detection of tiny pedestrians when viewed from bird's eye cameras.

Data preprocessing. Visual DL tools require data preparation, labeling, and augmentation. The COSMOS pilot node contains low-elevation cameras and high-elevation bird's eye view cameras, each requiring different type of preprocessing (Figure 2). The variation in angles and distances to the intersection, scale of objects, and overlapping field-of-views allow experimentation with the best view for a given application. For example, street-level cameras are closer to traffic objects. They consequently provide more visual details for applications such as multicamera object reidentification but are not as well suited to analyze large-scale traffic patterns due to the scale distortion among objects at varying distances to the camera—the bird's eye view cameras offer a better perspective for this type of application.

High-elevation cameras allow us to perform calibration transforms to improve the effectiveness of DL models. See in Figure 2 and Figure 3 that the high-elevation camera view can be adjusted to appear perpendicular to the road by applying a homography transformation, after which resizing and cropping of the frame create the square aspect ratio required by many DL models. In our traffic intersection use case, there are locations in the frame where relevant objects do not appear (that is, no cars on building walls or pedestrians flying in the air). This motivates the creation of (black) masks overlayed on top of the frames, as seen in Figure 3 and Figure 4.

Supervised object detection and tracking models require a large number of precisely annotated ground truth labels to train the algorithms. Producing accurate and consistent sets of labeled videos is difficult since both domain knowledge and significant amounts of time are needed. To support our experiments, we annotated thousands of frames capturing the intersection in various weather, lighting, and congestion conditions.

Object detection models typically struggle with small object detection. Tiny pedestrians in the bird's eye camera view, as well as far-away license plates in the street-level camera view, convey very little information. This results in relatively poor detection and tracking accuracies. To improve the performance, we have deployed techniques of training the DL models with a small-object drone acquired data set[7] and our COSMOS data sets and applying data augmentation techniques such as the copy/paste method illustrated in Figure 4(d).

Object detection and tracking. In smart traffic intersections, detecting pedestrians and vehicles and tracking their trajectories are the prerequisites for all downstream applications (Figure 5). This involves two computer vision tasks: object detection and MOT. The objective of object detection is to localize and classify objects within the frame. MOT aims to associate object identities across successive frames. State-of-the-art methods rely on DL blocks such as convolutional neural networks (CNNs)[8] and vision transformers.[9] These methods bring heavy computational cost, and the accuracy-speed tradeoff (the budgeting between computational complexity and inference speed) is vital to the success of smart city applications. With this consideration in mind, we experimented with a series of algorithms for detecting and tracking objects to find the best approach[10] based on our custom annotated data set for bird's eye videos. We choose YOLOv4[11] as the base detector for all downstream applications since it is able to provide accurate results in real time. Object detection performance is shown in Table 1, where the average precision (AP) and mean AP (mAP) are used as the evaluation metrics (Figure 6). On our bird's eye view intersection data, YOLOv4 outperforms both RetinaNet[12] and single-shot multibox detector (SSD)[13] in terms of AP and inference speed, where inference speed

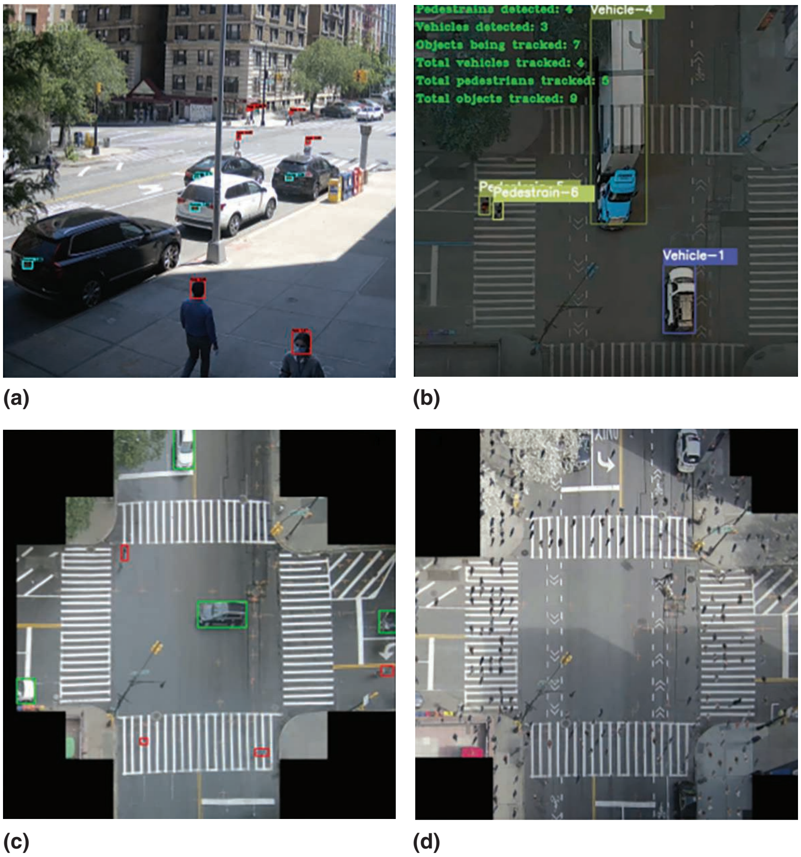

FIGURE 4. (a) YOLOv4 detections of faces and license plates in street level video. (b) SORT tracking of vehicles and pedestrians in bird's eye video. (c) Bird's eye ground truth bounding box labels of intersection objects. (d) Pedestrian copy-paste data augmentation for improving detection of small objects.

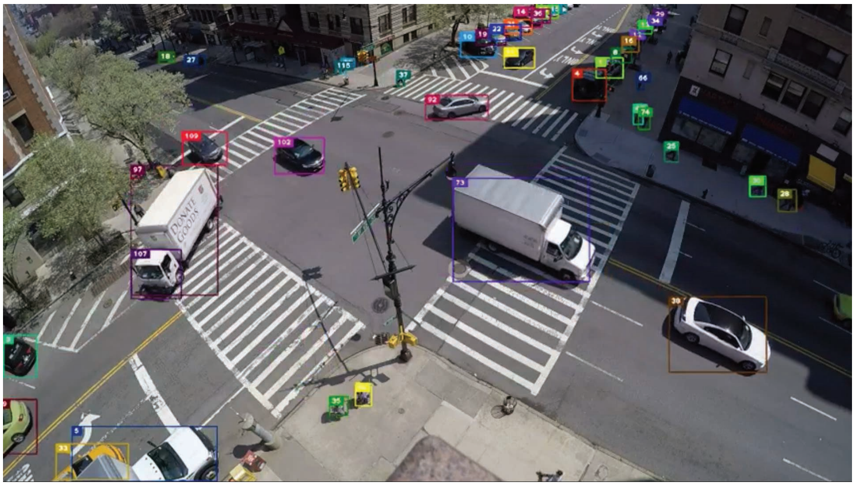

FIGURE 5. Pedestrian and vehicle detection on 120th Street and Amsterdam Avenue, fourth floor view.

TABLE 1. Object detection performance.

Model	Pedestrian AP (%)	Vehicle AP (%)	mAP (%)	Inference Speed*
YOLOv4	**66.31**	**97.58**	**81.95**	**34.99**
SSD	57.04	94.81	75.93	11.31
RetinaNet	20.83	95.59	58.21	22.97

*Inference speed (fps) on NVIDIA T4 GPU.

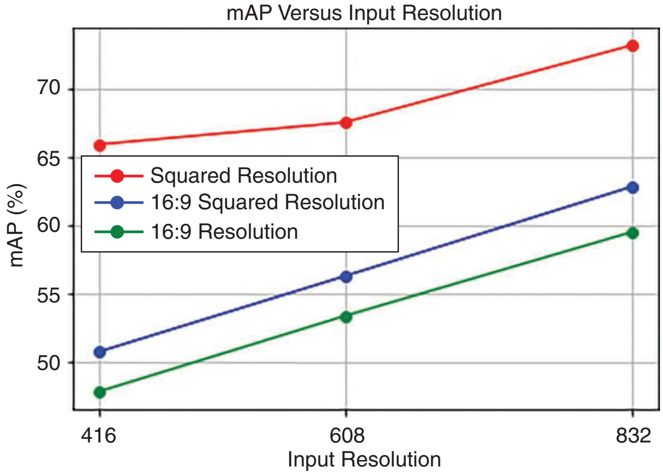

FIGURE 6. The mAP for pedestrians and vehicles; nine cases of image resolution versus aspect ratio.

is measured as the average time for a forward pass through the model with batch size equal to 1. For MOT, different scenarios need to be considered separately. For bird's eye cameras, object occlusions barely occur, so reidentification (reID) calculation is not as necessary as for the ground-level cameras. The reID calculation is often the computation bottleneck in MOT algorithms. "Simple online and real-time tracking" (SORT) and "simple online and real-time tracking with a deep association metric" (DeepSORT) suffice for the bird's eye view cameras. Illustrations for detection are shown in Figure 4.

Image resolution and object density. Highly elevated bird's eye cameras have a good view of the overall scene, shown in Figure 2. Pedestrians, which appear small, become a problem for object detection and tracking. Intuitively, the higher the resolution of the input image, the more object features can be preserved. However, higher resolution leads to a larger computational cost, thus making the inference slow. We tested a dozen combinations of image input resolutions and aspect ratios to find the best balance between accuracy and speed, three of which are shown in Figure 3. Some DL models, like YOLOv4,[11] perform better on input images with a fixed-sized, square aspect ratio. To maximize the preservation of important features of the intersection scene and to minimize the irrelevant components, the experiments indicate that the "squared cropped" 832 × 832 input produces the best results.[14]

Object density refers to the number of objects in a scene, which may impact the speed of inference as the busyness of the streets change through the day. We explored the inference time for ten 90-s videos where the number of objects varied from 4,000 to 26,000. The results show approximately 40% increase in computational load from the lowest to the highest density case. This indicates that object density measure can be used to switch among computational resources to obtain the optimal power/accuracy balance.

APPLICATIONS

Advances in video-based object detection and tracking have enabled the deployment of a number of traffic intersection applications, where one can identify the locations of objects in the intersection and classify them by type of vehicle, pedestrian, bicycle, and so on They can be tracked as unique entities which persist through the duration of traffic cycles, different camera views, and times of the day, week, month, and year. The abundance of spatial, temporal, and visual data makes it possible to perform data anonymization, quantization of traffic trends, crowd behavior surveillance, real-time intersection radar mapping, and more.

Privacy protection—face and license plate anonymization

Collecting real-time images and videos of public spaces from street level inadvertently involves capturing sensitive information such as faces and license plates. To avoid leaking private information with our data sets, we generated a pipeline to automatically blur these sensitive areas.[15] We trained several object detection models on a custom-labeled data set to detect faces and licenses for subsequent anonymization. When training with sequential video data sets, it is important to leave entire videos out of the training process to use for validation. Stationary objects—parked cars, seated pedestrians, chained bicycles, and so on—occur identically in many frames, and model evaluation on these stationary objects yields biased results. This leads to model overfitting and poor generalization to new intersection scenes, which has to be addressed.

Figure 7 shows an example input and output frame of the anonymization pipeline. For our face and license detection model, we chose YOLOv4[11] for its compromise between detection accuracy and inference speed. For privacy-critical applications, the most relevant performance measure is recall, the number of relevant faces and licenses that are detected out of the total number that pass through the frame. False positives are less of an issue than false negatives, as they result in an extra blurred area of the frame, but not a privacy leak. In our case, not all faces and licenses are "relevant"—some are too far away and too low resolution to be identifiable. We exclude these instances from the recall evaluation by defining pixel area thresholds below which the objects are ignored. We found that, below certain thresholds, facial features and license plate characters could not be reliably identified. While there exist information reconstruction techniques that could potentially recover these features, this is outside the scope of this project. Furthermore, we would need to reconsider our choice of anonymization as any form of blurring becomes ineffective. In the visible object evaluation, our pipeline blurs over 99% of visible faces and licenses and in the total evaluation it blurs over 96% of objects greater than 100 pixels.

To increase our confidence in the anonymization pipeline, we performed manual evaluations by inspecting anonymized output videos for misses, where a miss is defined as an object with more than a quarter of the face or license plate exposed. The results of the manual evaluations confirmed the results of the programmatic evaluations and shed some light on edge cases where our models consistently missed (Figure 8). Most edge cases were due to occlusions, such as occluded borders of license plates, pedestrian body occlusion, and tree branch occlusion, resulting in consistent false negatives. More data collection and training is needed to rectify these edge cases.

Counting objects

An important goal for smart intersections is to analyze traffic flow in real time. To this end, we use detection and tracking to classify and count vehicles and pedestrians and follow their paths through the intersection. Accumulation of the tracks provides sufficient data for traffic trend analyses that can be used to

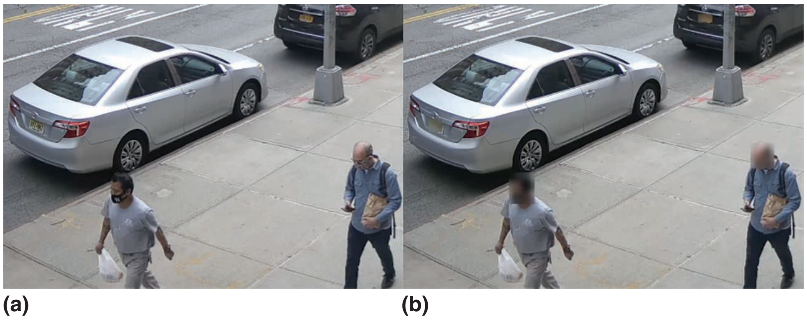

FIGURE 7. (a) The Input and (b) output of the face and license plate blurring pipeline.

optimize traffic flow and improve pedestrian safety in the intersection.

To perform object tracking, we use the detection-based (MOT) algorithm DeepSORT. DeepSORT requires an object detection model to provide the localization and classification information. Given detections of vehicles and pedestrians, DeepSORT uses a Kalman filter and Hungarian algorithm to map detections with similar sizes and motions across frames of a video. In this way, we can assign IDs to detected objects that persist throughout multiple video frames. Additionally, DeepSORT uses visual features of the object to increase the reliability of the tracking. Even if the object is not detected in consecutive frames, it can be assigned to the correct track by the reidentification model (reID) based on its visual features.

Though DeepSORT is a robust tracking system, it is still dependent on high-quality object detection. If an object is not detected or misclassified for multiple consecutive frames, it will be regarded by the algorithm as a "new track"—the old track disappears and a new one is created upon redetection. For vehicles, we achieve consistent high accuracy detection and corresponding high accuracy tracking, but for pedestrians, which have 4–5x smaller cross sections, high accuracy detection is a more challenging task. Pedestrian tracking accuracy suffers as a result of lower accuracy pedestrian detection. Data augmentation techniques, such as the copy-paste pedestrian method shown in Figure 4(d) and pretraining object detectors on small-object data sets show improvement for small-object detection, but pedestrian detection and tracking accuracies are still lower than for vehicles, with MOT accuracies (MOTA) of 75.16% and 18.23% for vehicles and pedestrians, respectively.

The vehicle tracking performance is sufficient for applications that quantify traffic flow. For example, in an automatic counting task we record vehicles passing through the intersection as turning right, turning left, or going straight from all four directions with an accuracy of 95% evaluated over 21 min of a video recording.

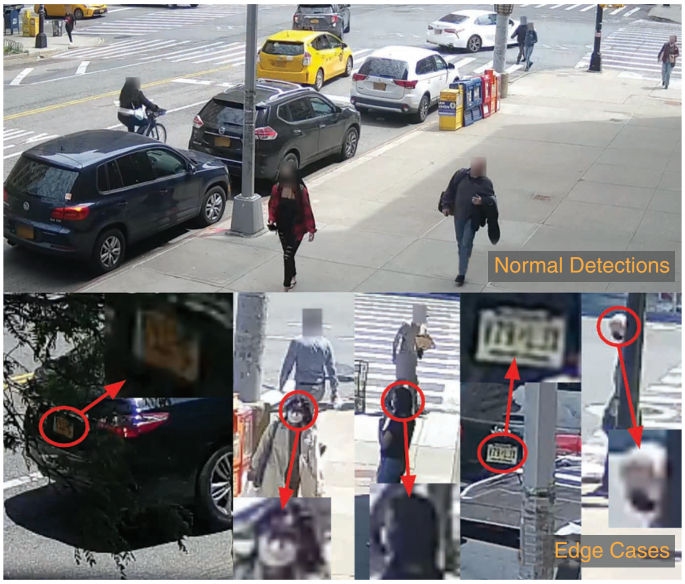

FIGURE 8. Successful blurring detections (top) versus edge cases (bottom).

Social distancing in pandemics

Smart cities can assist in combating global pandemics, such as COVID-19, by providing means for monitoring, analyzing, and potentially controlling social distancing behavior. We proposed several techniques and applied them to video data sets collected at the COSMOS pilot intersection.

The fundamental idea is to estimate distances between pedestrians and compare them against the recommended minimal distance threshold. The first step is to detect the pedestrians. The real-world distance is then estimated by calculating the pixel-wise distance between pedestrians within one frame. The tracking of pedestrians between frames facilitates the calculation of higher order statistics, related to safe social distancing groups, which are more meaningful than an individual-to-individual social distancing violation rates. When acquaintances are walking together on the street as a" safe group," the intragroup distance is often smaller than the social distancing threshold, which (incorrectly) triggers the indication of the violation. To solve this problem, we utilize the pedestrian trajectory similarity and stability, which can evaluate the motion dynamic between every pedestrian pair. This group validation approach is able to significantly reduce the number of false positive violations, achieving the F1 score of 0.92. Based on this approach, we built a social distancing analysis (SDA) system B-SDA[16] for bird's eye view cameras, as well as a complementary method Auto-SDA[17,18] with ground-level cameras.

An example of the results obtained with the bird's eye video data set (Figure 9) shows the distribution of the duration of social distancing violations during the Covid-19 pandemic. Figure 10 shows the social distancing violation rates for the street-level camera data set 1) during the pandemic and 2) after the vaccine was widely available. Detailed analyses and comparisons of multiple statistics before the pandemic and during the pandemic demonstrate that the proposed systems can reliably identify social distancing violations.

Real-time “radar screen”

The “radar-screen” application aims to infer positions and velocities of objects within a traffic intersection and broadcast them to the participants in the intersection in real time, as illustrated in Figure 11. The information can be distributed in raw or coded/meta format. The application intends to provide a real-time service with latency of 1/30 s between the observation of objects and the wireless broadcast delivery. As described previously, this is motivated by the approximation of a 10-cm vehicle movement with speed of 10 km/h. The application includes the acquisition of videos from surrounding buildings, potential harvesting of videos (or encoded data) from cameras within vehicles, harvesting of IoT sensor data, transmission via a high-speed network to the inference computer, data aggregation and preprocessing, DL-based object detection and tracking, extraction of information at a higher abstraction level, and (in a more advanced version) deduction of commands that may be issued to individual vehicles after optimizing the traffic flow. The final step is the broadcasting of information. This is an aspirational application in that achieving the cumulative latency of 33.3 ms is technologically very challenging. Balancing between computational capabilities, power consumption, and latency minimization of the extreme edge compute units or edge computing centers, requires rapid sensor data acquisition and dynamic network and resource control. This application motivates research to optimize each of the building blocks described in previous sections of this article as well latency-focused cross-module system integration.

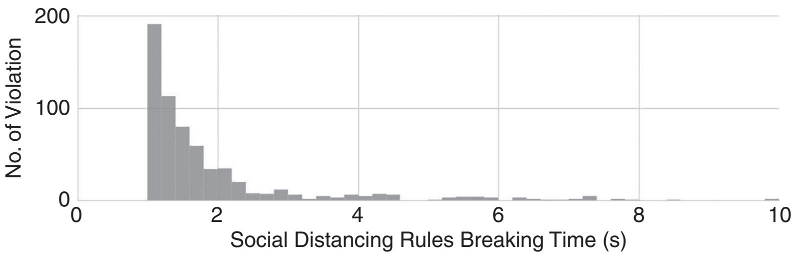

FIGURE 9. B-SDA: Distribution of the duration of social distancing violations.

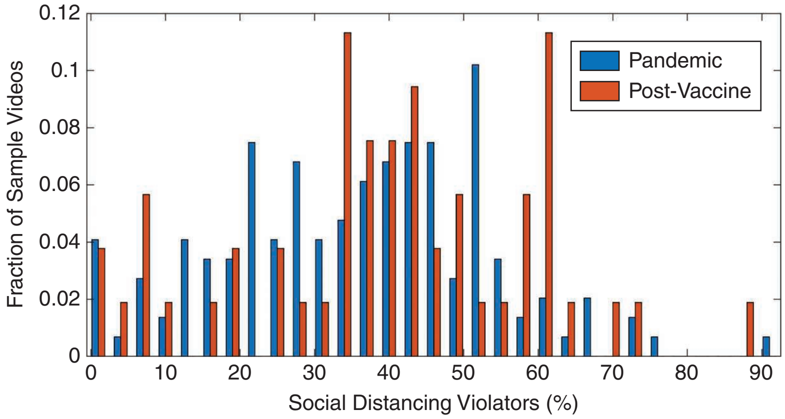

FIGURE 10. Auto-SDA: Normalized histogram of the percentage of social distancing violations.

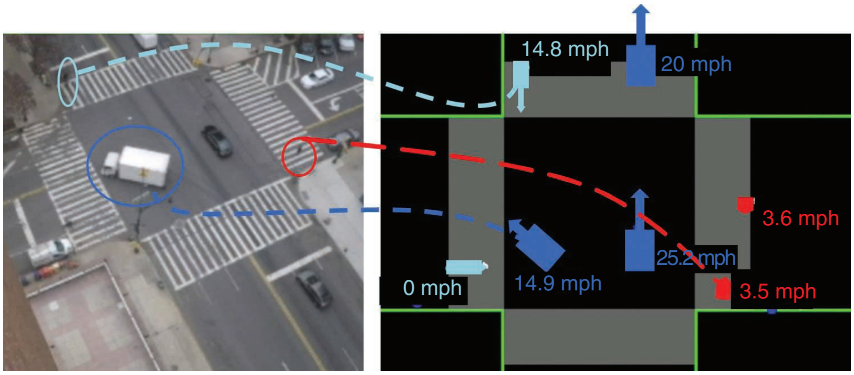

FIGURE 11. The “radar screen”: one frame of a video containing locations and velocities of objects within an intersection.

Traffic management

Intelligent nodes located at individual intersections provide powerful data acquisition and intelligent edge-computing. On a larger scale, smart cities require the aggregation of data from multiple intersections and mutual coordination. In that vein, we have commenced collaborative studies with traffic engineering experts on the definition of key parameters, such as timing resolution, sensor locations, and application programming interfaces for data exchange between intelligent smart intersection nodes and traffic optimization systems.[19] We are building simulators and defining digital twins that will play predictive roles in the behavior of individual traffic participants and in global optimization of traffic management.

A vision of the smart city intersection as the intelligence node for future metropolises has been presented. The proposed architecture is driven by societal needs to preserve privacy, which strongly implicate edge computing and intelligence as the key paradigm for data management and processing. Key technological components have been reviewed, such as sensors, networks, and edge AI computing. Real-time needs of future safety-critical systems have been examined, and design considerations for the aspirational "radar-screen" application, which closes the loop from sensors to actuators, have been summarized. The requirements for low latency, based on the 33.3-ms target, have been explored. System integration challenges have been illustrated using the examples from experiments performed on the pilot node of the COSMOS testbed in New York City.

Our research points to the following exploration topics:

1. State of the art DL-based object detection models are comprised of over 60 million parameters and require passing more than 100 convolutional layers, where each convolution has complexity. Model optimization techniques like weight pruning, inference scheduling, and neural algorithmic search strategies[20] need to be incorporated into practical systems.
2. Reliance on supervised data sets for video processing is not scalable due to the labeling cost and quality concerns. This necessitates research on unsupervised learning methodologies which should be based on continuous or active learning and take advantage of the peculiarities of the fixed scene within a traffic intersection.[21]
3. Data fusion from multiple cameras is expected to yield notable improvements in detection and tracking accuracies.
4. Achieving low latency for low-rate little-data applications is possible by using processing on the "extreme edge," but meeting the requirements of 1/30-s latency for high-resolution videos is a challenge. New video coding methods and streaming protocols should be explored with focus on localized low-latency performance.

ACKNOWLEDGMENT

This work was supported in part by National Science Foundation grants CNS-1827923, CNS-1910757, OAC-2029295, CNS-2038984, CNS-2148128, EEC-2133516, and AT&T VURI award.

REFERENCES

1. L. Sánchez et al., "SmartSantander: IoT experimentation over a smart city testbed," *Comput. Netw.*, vol. 61, pp. 217–238, Mar. 2014, doi: 10.1016/j.bjp.2013.12.020.
2. L. Belli et al., "IoT-enabled smart sustainable cities: Challenges and approaches," *Smart Cities*, vol. 3, no. 3, pp. 1039–1071, Sep. 2020, doi: 10.3390/smartcities3030052.
3. A. Y. Ding et al., "Roadmap for edge AI: A Dagstuhl perspective," *SIGCOMM Comput. Commun. Rev.*, vol. 52, no. 1, pp. 28–33, Mar. 2022, doi: 10.1145/3523230.3523235.
4. D. Raychaudhuri et al., "Challenge: COSMOS: A city-scale programmable testbed for experimentation with advanced wireless," in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw.*, Apr. 2020, pp. 1–13, doi: 10.1145/3372224.3380891.
5. P. Skrimponis et al., "COSMOS educational toolkit: Using experimental wireless networking to enhance middle/high school stem education," *SIGCOMM Comput. Commun. Rev.*, vol. 50, no. 4, pp. 58–65, Oct. 2020, doi: 10.1145/3431832.3431839.
6. X. Xu, Q. Huang, X. Yin, M. Abbasi, M. R. Khosravi, and L. Qi, "Intelligent offloading for collaborative smart city services in edge computing," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 7919–7927, Sep. 2020, doi: 10.1109/JIOT.2020.3000871.
7. P. Zhu et al., "VisDrone-DET2018: The vision meets drone object detection in image challenge results," in *Proc. ECCV Workshops*, 2018, pp. 437–468.
8. L. Liu et al., "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, Feb. 2020, doi: 10.1007/s11263-019-01247-4.
9. A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

ABOUT THE AUTHORS

ZORAN KOSTIĆ is a professor of professional practice at Columbia University, New York, NY 10027 USA. His research interests include the Internet of Things, physical data analytics, and applications of deep learning in smart cities, medicine, and health. Kostić received a Ph.D. in electrical engineering from the University of Rochester. Contact him at zk2172@columbia.edu.

ALEX ANGUS is at Qualcomm, Inc., San Diego, CA 92121 USA. His interests include applications of deep learning in smart cities and parallel computing architectures. Angus received an M.S. in electrical engineering from Columbia University. Contact him at alexsugna@gmail.com.

ZHENGYE YANG is a doctoral student in electrical engineering at Rensselaer Polytechnic Institute, Troy, NY 12180 USA. His research interests include computer vision, multimodal learning, and deep learning. Yang received an M.S. in electrical engineering from Columbia University. Contact him at yangz15@rpi.edu.

ZHUOXU DUAN is a Ph.D. student at Rensselaer Polytechnic Institute, Troy, NY 121180 USA. His research interests include computer vision, deep learning, natural language processing, and multimodal understanding. Duan received an M.S. in electrical engineering from Columbia University. Contact him at duanz2@rpi.edu.

IVAN SESKAR is the chief technologist at WINLAB, Rutgers University, North Brunswick, NJ 08902 USA. His research interests include wireless communications, large-scale networking research testbeds, and applications. Seskar received an M.S. in electrical engineering. He is a Senior Member of IEEE and the cochair of the IEEE Future Networks Testbed Working Group. Contact him at seskar@winlab.rutgers.edu.

GIL ZUSSMAN is a professor of electrical engineering and computer science at Columbia University, New York, NY 10027 USA. Zussman received a Ph.D. in electrical engineering from the Technion. He is a Fellow of IEEE. Contact him at gil@ee.columbia.edu.

DIPANKAR RAYCHAUDHURI is at WINLAB, Rutgers University, North Brunswick, NJ 08902 USA. He is the PI for the COSMOS testbed. His research interests include future network architectures and protocols, wireless systems and technology, dynamic spectrum access and cognitive radio, and experimental prototyping and network research testbeds. Raychaudhuri received a Ph.D. from the State University of New York at Stony Brook. He is a Fellow of IEEE. Contact him at ray@winlab.rutgers.edu.

10. S. Yang et al., “COSMOS smart intersection: Edge compute and communications for bird’s eye object tracking,” in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (SmartEdge)*, 2020, pp. 1–7, doi: 10.1109/PerComWorkshops48775.2020.9156225.
11. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” 2020, *arXiv:2004.10934*.
12. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2999–3007, doi: 10.1109/ICCV.2017.324.
13. W. Liu et al., “SSD: Single shot multibox detector,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
14. Z. Duan et al., “Smart city traffic intersection: Impact of video quality and scene complexity on precision and inference,” in *Proc. IEEE Smart City’21*, 2021, pp. 1521–1528, doi: 10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00226.
15. A. Angus, Z. Duan, G. Zussman, and Z. Kostic, “Real-time video anonymization in smart city intersections,” in *Proc. 2022 IEEE 19th Int. Conf. Mobile Ad Hoc Smart Syst. (MASS)*, to be published.
16. Z. Yang, M. Sun, H. Ye, Z. Xiong, G. Zussman, and Z. Kostic, “Bird’s-eye view social distancing analysis system,” in *Proc. 2022 IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, pp. 427–432, doi: 10.1109/ICCWorkshops53468.2022.9814627.
17. M. Ghasemi et al., “Demo: Video-based social distancing evaluation in the COSMOS testbed pilot site,” in *Proc. ACM MOBICOM’21*, 2021, pp. 1–3, doi: 10.1145/3447993.3510590.
18. M. Ghasemi, Z. Kostic, J. Ghaderi, and G. Zussman, “Auto-SDA: Automated video-based social distancing analyzer,” in *Proc. ACM HotEdgeVideo*, 2021, pp. 7–12.
19. G. Karagiannis et al., “Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions,” *IEEE Commun. Surveys Tuts.*, vol. 13, no. 4, pp. 584–616, Fourth Quarter 2011, doi: 10.1109/SURV.2011.061411.00019.
20. C. R. Banbury et al., “Benchmarking tinyML systems: Challenges and direction,” 2020, *arXiv:2003.04821*.
21. Z. Dai, G. Wang, S. Zhu, W. Yuan, and P. Tan, “Cluster contrast for unsupervised person re-identification,” 2021, *arXiv:2103.11568*.