Increasing Traffic Safety with Real-Time Edge Analytics and 5G

Ivan Lujic
Vienna University of Technology
ivan@ec.tuwien.ac.at

Vincenzo De Maio
Vienna University of Technology
vincenzo@ec.tuwien.ac.at

Klaus Pollhammer
Swarco Futurit
klaus.pollhammer@swarco.com

Ivan Bodrozic
Vienna University of Technology
ivan.bodrozic.00@fesb.hr

Josip Lasic
Vienna University of Technology
jlasic00@fesb.hr

Ivona Brandic
Vienna University of Technology
ivona@ec.tuwien.ac.at

ABSTRACT

Despite advances in vehicle technology and road modernization, traffic accidents are a huge global issue, causing deaths and injuries, especially among pedestrians and cyclists. This often happens due to pedestrians and cyclists in drivers' blind spots or distractions delaying drivers' reactions. Therefore, timely warning drivers about critical situations is important to increase traffic safety. New edge computing and communication technologies have been proposed to reduce latency in critical IoT systems. However, state-of-the-art solutions either do not focus on traffic safety or do not consider low-latency requirements in this context.

We propose InTraSafEd5G (Increasing Traffic Safety with Edge and 5G) to address these issues. InTraSafEd5G performs real-time edge analytics to detect critical situations and deliver early warnings to drivers. After describing our design choices, we provide a prototype implementation and evaluate its performance in a real-world setup. The evaluation shows that InTraSafEd5G can (i) detect critical situations in real-time and (ii) notify affected drivers in 108.73*ms* on average using 5G, which is within expected latency requirements of road safety IoT applications. Our solution shows a promising step towards increasing overall traffic safety and supporting decision-making in critical situations.

CCS CONCEPTS

• **Computer systems organization → Sensor networks**; • **Hardware → Sensor applications and deployments**; • **Information systems → Data analytics**.

KEYWORDS

edge computing, data analytics, real-time, computer vision

1 INTRODUCTION

Ensuring road traffic safety represents an important challenge [6]. According to [1], at least 51300 pedestrians and 19450 cyclists were killed on EU roads between 2010 and 2018, accounting deaths among pedestrians and cyclists for 29% of all EU road deaths, as reported by the European Transport Safety Council (ETSC). Causes of accidents involve distractions and poor visibility. Early warning systems can improve traffic safety by promptly notifying drivers about critical situations (e.g., pedestrians or cyclists in drivers' blind spots [6]), allowing them to take actions to avoid accidents [16]. However, since drivers' brake reaction time is measured around 1500ms on average [7], early warning systems must detect and send timely notifications respecting strict latency constraints.

Recently, edge computing has been proposed as a solution to reduce latency in critical systems. However, several challenges need to be addressed to allow drivers to receive notifications within strict latency constraints. Challenges are related to (i) placement of edge nodes respecting urbanistic space constraints; (ii) selection of efficient detection algorithms to identify dangerous situations within given constraints; (iii) selection of a network technology allowing fast aggregation and delivery of information to drivers; (iv) providing real-time notifications to multiple vehicle drivers approaching the critical intersections.

Existing work in this regard targets vehicles with V2X communication capabilities [14]. Other works either do not take into account pedestrians' safety [17] or do not consider strict latency constraints [5]. Some modern vehicles provide pedestrian detection feature and automated emergency braking (i.e., automatically applying the vehicles' brakes to assist drivers in preventing a crash). However, modern vehicles with advanced technologies can be more expensive and inaccessible to most road users.

We propose InTraSafEd5G (<u>In</u>creasing <u>Tra</u>ffic <u>Saf</u>ety with <u>Ed</u>ge and <u>5G</u>), a system for detecting pedestrians and cyclists in drivers' blind spots at critical traffic intersections and reporting their presence to drivers. Detection of pedestrians is performed by applying an object detection algorithm on video frames streamed from cameras that are attached to edge devices, and integrated on traffic lights. Detection results are delivered in real-time using low-latency 5G communication. InTraSafEd5G is designed to provide audio and visual notifications on the drivers' mobile phones, thereby providing early warnings about critical situations in the driver's path.

First, we present the conceptual design of InTraSafEd5G and motivate our design choices in Section 2. Then, we provide a prototype implementation of InTraSafEd5G and deploy it on a critical intersection in the city of Vienna, as described in Section 3. We evaluate InTraSafEd5G performance in Section 4, showing its capability to detect critical situations and deliver early warnings in the order of tens to hundreds of milliseconds [13]. Finally, we describe related work in Section 5 and conclude our paper in Section 6.

2 BACKGROUND

2.1 Motivational Scenario

In the last 10 years, the city of Vienna has experienced around 50000 casualties from traffic accidents, resulting in more than 100 deaths [1]. Many casualties are caused due to blind spots in the drivers' sight or distractions while driving, causing inabilities to brake on time. Figure 1 illustrates our motivational scenario. We consider a typical

[1]https://www.wien.gv.at/english/administration/statistics/traffic-accidents.html

example of a crosswalk situation on the intersections, in which both drivers and users (e.g., pedestrians, cyclists) can not detect potentially threatening situations on time. Such situations can include roadside obstacles and conditions that can reduce traffic safety such as (i) roadside bus stations and buses covering cyclists approaching the intersection, (ii) bushes and trees hiding children and animals that are moving towards the crosswalk (e.g., a child running after a dog, out of parents' sight), (iii) bad weather conditions causing poor visibility (e.g., fog, heavy rain). Also, pedestrian-vehicle conflicts can often happen due to unexpected behaviors of pedestrians near crosswalks, such as sudden acceleration and deceleration [2].

To increase traffic safety and support drivers and users on critical intersections to avoid accidents, we consider (i) exploiting modern AI and computer vision techniques to detect critical situations and, (ii) exploring possibilities of edge computing paradigm and emerging 5G network connection. We design our traffic safety solution in the scope of InTraSafEd5G project, funded by the city of Vienna to explore 5G use cases for a better connected smart city.

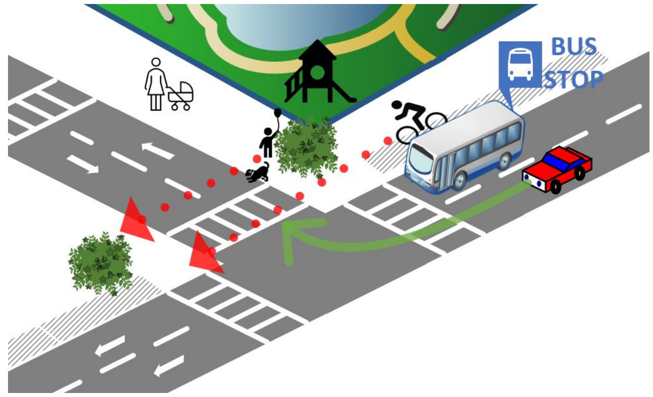


Figure 1: An example traffic scenario to illustrate the critical situations when pedestrians and cyclists appear in the driver's blind spots (e.g., behind roadside bus station, bushes).

2.2 Requirements

We identified a specific intersection in the city of Vienna that is considered as critical for traffic accidents, due to roadside obstacles and blind spots. Swarco Futurit, a Vienna traffic infrastructure provider, enabled access to empty traffic lights chambers, in which the corresponding edge hardware is installed. We consider the following properties and requirements in the InTraSafEd5G design:

- *Low latency*: Low latency is an important requirement to increase traffic safety, since delivering timely notifications to drivers is important to avoid accidents. Total latency is given by the sum of (i) computation time, (ii) communication (data transfer or message) overhead, (iii) reliability overhead. The sum of these values must respect the drivers' average brake reaction time [7], therefore notification latency must be ranging in the order of tens to hundreds of milliseconds [13].
- *Privacy preservation*: Many edge sensors and devices can collect sensitive data about people (e.g., people's faces). In this context, any network transfer of sensitive data should be minimized or anonymized (e.g., through aggregation, numbers). Compared to traditional scenarios of sending sensitive data over the network to the cloud, we should ensure that no sensitive data are stored or sent over the network [4].
- *Space limitations*: Edge-relevant hardware must be deployed close to critical intersections, respecting urbanistic space constraints. In our setup, the city of Vienna infrastructure provider provided access to empty traffic light chambers for the deployment of edge hardware. Provided traffic lights chambers can be found in most of the traffic infrastructures worldwide [15].
- *Low cost*: Edge hardware (e.g., AI device, camera) has to be installed on many crossroads of a metropolitan area. For this reason, hardware costs must be contained. In our use case, the traffic infrastructure provider of the city of Vienna estimated the costs for a single edge node installation to be in the range of a few hundreds of euros.

3 SYSTEM DESIGN

Figure 2 shows an overview of the proposed architecture design, based on the challenges and requirements identified in Section 2. We illustrate two main components:

Computation component is responsible for the real-time processing of data coming from edge sensing devices. It includes both hardware and software parts. The hardware part includes adapted configurations of edge devices, considering space requirements, and plugged cameras to constantly monitor critical intersections and crosswalks. The software part includes edge processing, which processes video frames using deep learning object detection modules. Object detection workflows are based on a neural network, trained to detect pedestrians and cyclists.

Communication component is responsible for real-time delivery of the edge processing output that is critical for early warnings of drivers. It includes both network and application parts. The network part includes a mechanism to efficiently and timely broadcast such information to a multiple and dynamic number of vehicle drivers approaching the intersection. The application part includes implemented modules for tracking the vehicle movements as well as the application interface with warning features.

In Step 1, vulnerable road users (e.g., pedestrians and cyclists) are captured in video frames collected by a deployed traffic camera. Input video frames are constantly analyzed by the edge processing module in Step 2, searching for target users in the critical crosswalk or intersection area. Once users approaching this monitored area are detected (Step 3), a notification message is generated in Step 4. The message is then forwarded to an app installed on the driver's mobile device, which shows visual and audio notification (Step 5).

Regarding the privacy property, the video frames, captured by cameras, are analyzed immediately at the edge device, ensuring that sensitive data are neither transferred nor stored over the network. Only analytics output, i.e., a number of objects detected (i.e., pedestrians, cyclists, or animals), is transmitted to drivers' mobile devices. Technology evaluation is described in the following sections.

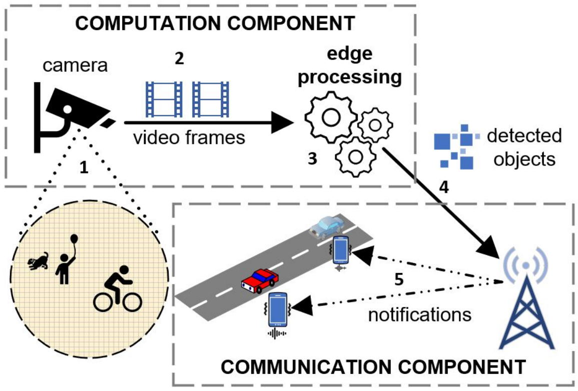


Figure 2: Architecture overview.

3.1 Computation Component

3.1.1 Hardware. Object detection performance depends on selected hardware. Due to physical space limitations of the traffic light chamber, we employ single-board Raspberry Pi (RPi) edge devices, to which we attach co-processors to improve performance of neural network inference. As co-processors, we evaluate Google's Coral Edge TPU accelerator, and Intel's Neural Compute Stick 2 (NCS2) since both can be plugged via USB and used for vision-based ML applications. We select Coral Edge TPU, since it supports TensorFlow Lite models (lightweight solution to run ML TensorFlow models on edge devices). Considering space and low-latency requirements, we evaluate different RPi models with/without Coral's Edge TPU. Table 1 shows technical details. To capture video frames from the target intersection, we use RPi 8MP Camera Module V2.

3.1.2 Software. Software module is developed using TensorFlow Lite, a version of popular TensorFlow framework optimized for limited IoT devices, including RPi. To select the best performing edge configuration, we first collected video frames from the chosen intersection, on which the prototype should be deployed. Then, we evaluated the performance of both quantized MobileNet SSD v1 and v2, lightweight and pre-trained convolutional neural network (CNN) based object detection models, trained using the standard COCO [10] dataset, on our collected dataset. Figure 3 shows the average inference time per single frame on different edge node configurations. The results are averaged over 100 frames for statistical significance, since by adding more frames the difference in standard deviation of inference times is below 39 μs on average. We observe in Table 1 that RPi 4 with Edge TPU has overall the lowest inference time for both models.

Furthermore, we check confidence scores of the model, calculated using Tensorflow confidence function[2], for both models in Figure 4. We consider quantized model versions, to further improve latency with limited effect on inference score. Here, the object detection module is set to only detect a class "person" from a collected dataset with a confidence threshold of 0.5 (i.e., a cut-off threshold for accepting detection results). Although MobileNet SSD v1 has a

[2]https://www.tensorflow.org/lite/models/object_detection/overview

Table 1: Edge node configurations.

Node type	CPU	RAM [GB]	Edge TPU
RPi 3 B+	Quad-core Cortex-A53 (ARMv7) at 1.4GHz	1	no
RPi 3 B+	Quad-core Cortex-A53 (ARMv7) at 1.4GHz	1	yes
RPi 4	Quad-core Cortex-A72 (ARMv7) at 1.5GHz	4	no
RPi 4	Quad-core Cortex-A72 (ARMv7) at 1.5GHz	4	yes

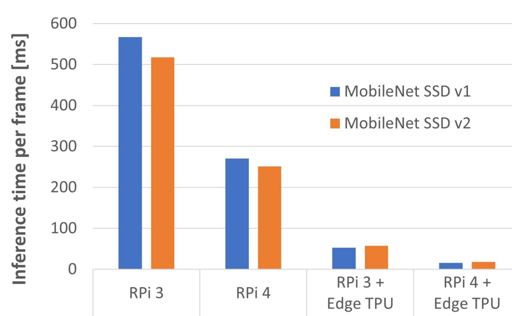
slightly lower inference time by 2.46ms (or 13.84%) on average, MobileNet SSD v2 shows a better inference score of 16.12% on average. Thus, we select as the best option MobileNet SSD v2 running on RPi 4 with Edge TPU.

3.2 Communication Component

InTraSafEd5G is designed to work in the mobile context, thus we consider 3G, 4G and 5G physical and transport layers. We describe now all design choices related to the communication component.

3.2.1 Network Protocol. Since the main goal of InTraSafEd5G is to provide notifications of critical situations within specific time frames, selected communication protocol should (i) have a minimal message overhead to keep the transfer time and latency low, (ii) offer guaranteed delivery of messages to users, and (iii) avoid unnecessary network flooding. Traditional client-server communication (e.g., HTTP in cloud-based web applications), is not suited for this scenario, due to high message overhead and the necessity of polling to be notified about new events (e.g., a pedestrian detected in a blind spot). We focus then on publish/subscribe (Pub/Sub) protocols, which allow event-based notification and dynamical targeting of drivers close to a particular crossroad.

We select the following protocols for evaluation, used in different contexts requiring near real-time latency: CoAP, DDS, AMQP, MQTT. In Table 2 we show the result of our comparative study. First, we focus on Pub/Sub protocols, to enable event-triggered data transmission and avoid unnecessary polling. Also, since accurate detection of pedestrians and cyclists in our scenario needs to aggregate sensor data in a single processing point, a protocol designed for centralized processing is more desirable than a distributed protocol. Another desirable feature is the possibility to choose between different Quality of Service (QoS) levels according to provided latency.



	RPi 3	RPi 4	RPi 3 + Edge TPU	RPi 4 + Edge TPU
MobileNet SSD v1	566.84	270.34	52.41	15.32
MobileNet SSD v2	517.63	251.44	57.44	17.78

Figure 3: Inference time observation for different edge node configurations and two object detection models.

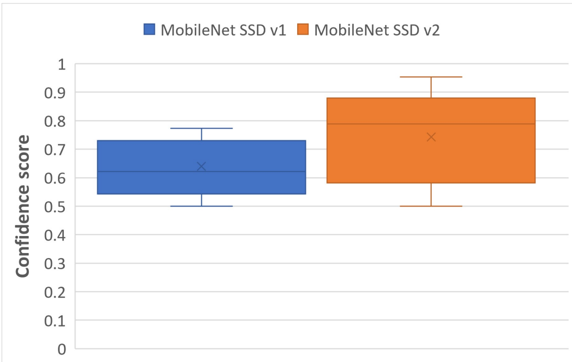


Figure 4: Confidence score observation for two object detection models on RPi 4 with Edge TPU (threshold set to 0.5).

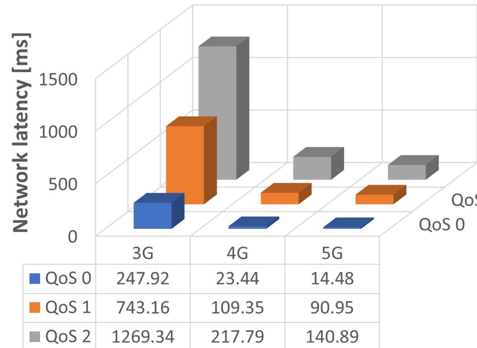


Figure 5: MQTT network latency for QoS=$\{0, 1, 2\}$ over different network types (3G, 4G and 5G).

A good level of security should also be provided, since no malicious users should be able to inject fake detection data and potentially causing troubles. Finally, we want to ensure that header size of each message is low, to reduce network load.

Based on our analysis, we do not select CoAP and AMQP since they do not offer a publish/subscribe communication model. Between publish/subscribe protocols (DDS and MQTT) we select MQTT, due to its lower overhead and the fact that it is most suited for centralized processing.

3.2.2 Quality of Service. Besides strict low-latency of delivery, InTraSafEd communication layer has also to provide guarantees on delivery of the notification. Indeed, potential effects of unreliable communication might cause the loss of important updates, with negative effects on traffic safety. Since our communication layer is based on the MQTT protocol, we focus on MQTT QoS layers.

MQTT offers three different QoS levels: QoS 0, where messages are delivered AT MOST ONCE (no guarantee on message delivery); QoS 1, where messages are delivered AT LEAST ONCE (message delivery is guaranteed, but replications might occur), and QoS 2, where messages are delivered EXACTLY ONCE. Since selected QoS affects latency of notifications, we measure the latency of MQTT messages in our scenario using different QoS levels.

Latency measurements are performed employing a self-developed Android mobile app, which first subscribes to a specific topic, then it sends a message with a unique id using the same topic. Once it receives the same messages, it calculates the latency based on the round-trip time of the message. The payload of the message is set to 64 bytes that is realistic for the amount of data managed by the application (the number of pedestrians and cyclists in a blind spot).

Table 2: Comparative study of IoT protocols. Pub/Sub = Publish/Subscribe; R/R = Request/Reply; P2P = Point to Point.

Protocol	Paradigm	Processing	QoS levels	Security	Header size [bytes]
CoAP	R/R	Distributed	4	DTLS	4
DDS	Pub/Sub	Distributed	5	TLS	48
AMQP	P2P	Impl. specific	3	TLS	8
MQTT	Pub/Sub	Centralized	3	TLS/SSL	2

Figure 5 shows network latency offered by different QoS levels on different network layers (3G, 4G and 5G). Results are obtained by calculating the average latency of 100 messages, for statistical significance. From the plot, we see that in most of the cases average notification latency falls within the requirements of tens to hundreds of milliseconds for road safety applications [13]. However, when selecting QoS 2, the 95% confidence interval of our measurements for 3G includes values above 1000ms. For this reason, we select QoS 1, which ensures latency below 800ms in the worst case (3G, QoS 1). Also, in our scenario, average notification latency is 109.35ms and 90.95ms, respectively for 4G and 5G.

3.2.3 Protocol Setup. Finally, we describe the general MQTT protocol setup. MQTT message routing relies on a software component called MQTT broker, which receives messages published by different clients (publishers) and forwards them to clients subscribed to the messages' topic (subscribers). In the target scenario, each camera with RPi represents a publisher, and a mobile app is a subscriber. In the current prototype, topic subscriptions are set up when the application is started. In future work, we plan to investigate location-based subscriptions, as described by [8]. We select Mosquitto MQTT broker [9], which offers enough security and flexibility for our requirements.

Each client needs access to the broker over the Internet. Since latency is the main requirement of our application, we evaluate the latency of deploying the broker at the edge (inside TU Wien's infrastructure) or using a cloud service instead. Evaluation of latency of both deployments is shown in Figure 6. We can see that edge deployment significantly reduces the latency (up to 50.20% and 47.18% for 4G and 5G, respectively), making edge broker placement as the best choice for our latency-critical scenario.

3.3 Mobile Client

Notifications of critical situations are sent to a mobile client installed on the driver's mobile phone. InTraSafEd5G mobile client is developed for Android 10 using Kotlin. We leave implementation of iOS and Windows Phone versions for future work. The application works as follows: first, the app subscribes to the topic representing a monitored critical area and registers to the broker. Once the message is received, the app visualizes a message using overlays and

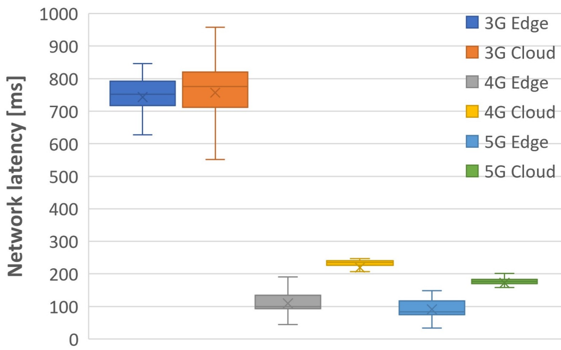
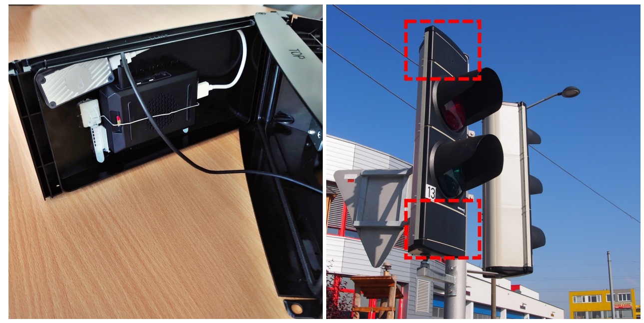
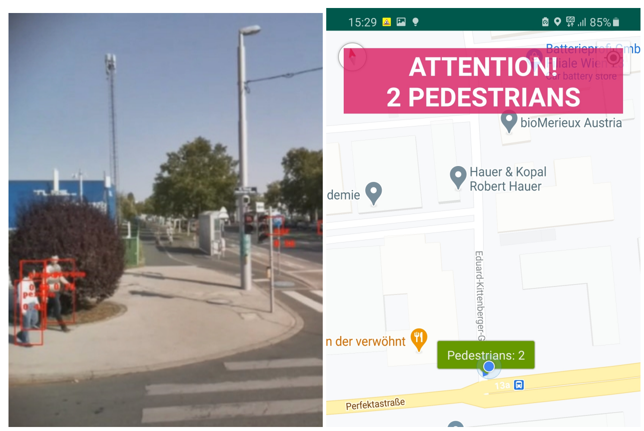


Figure 6: MQTT network latency for QoS=1 w.r.t. edge/cloud broker placement.



(a) Edge node deployment. **(b) Smart traffic light integration.**

Figure 7: Edge nodes setup on Vienna's chosen intersection and the integration into the traffic-signal chambers.



(a) Pedestrian detection. **(b) Screenshot of user application.**

Figure 8: Real-time edge analytics demonstration for increasing traffic safety. Subfigure (a) shows the output of object detection (during the project demonstration day) on edge node Raspberry Pi 4B, while Subfigure (b) shows the application running on the driver's Samsung S20 5G phone.

playing an audio notification. Communication with the MQTT broker is performed through the PAHO MQTT library 1.1 for Android, while the location is obtained using the mobile phone's GPS and Google Maps SDK v3.1.0. An example of the mobile client interface is shown in Figure 8b.

4 SYSTEM DEPLOYMENT

4.1 Design Choices

InTraSafEd5G collects video frames from cameras deployed on smart traffic lights, targeting drivers' blind spots. Video frames are then processed by an object detection algorithm, implemented using Python 3.7, to identify critical situations (e.g., pedestrians and cyclists outside drivers' field of vision). To this end, we (i) deployed edge nodes using RPi 4B, to which we attached a camera and Edge TPU to speed up object detection (Figure 7a) and (ii) set up a network connection using a HTC 5G Hub (5G network coverage is set exclusively for the prototype deployment). We placed the edge node (above) and 5G Hub (under) specific traffic lights (Figure 7b). Figure 8a illustrates the detection of pedestrians in drivers' blind spots (e.g., behind bushes or bus stations) by installed edge nodes. Results from different edge nodes are then aggregated and audio and visual notifications are sent to the driver's mobile device if a critical situation is detected while a driver approaches the covered intersection (Figure 8b).

4.2 Discussion

Transmission latency on 5G is measured to be around 90.95ms for QoS 1, which guarantees enough time for timely notifications. Regarding the computational performance on the edge device, object detection on camera-collected frames takes around 17.78ms using a lightweight RPi 4 with Edge TPU. This solution not only reduces cost while providing high-level performance, but also allows increased privacy, as data taken by cameras do not travel over the network nor need to be stored on the devices. Overall, in our InTraSafEd5G use case, detection of pedestrians on the crossroad together with the required latency to notify drivers results in 108.73ms on average. We also observed that the proposed system with QoS level 0 results in overall latency of 32.26ms on average for 5G. However, although this QoS level does not guarantee the delivery of notifications, it is appropriate in situations where (i) the connection is reliable (e.g., cellular antennas deployed close to the target area), and (ii) message loss on a small scale would not affect the early warning system (e.g., capturing and processing higher frame rates). Still, in future work we plan to investigate the scale of message loss using QoS level 0.

The installation of the proposed solution on a higher scale of a metropolitan area, would require (i) identification of critical intersections, (ii) installation of 5G network coverage, (iii) integration of the proposed design choices, and (iv) owning a 5G-enabled phone. Although using 5G connection can offer the full benefits of the proposed solution, (i) owning a 5G-enabled phone can be costly for users, as well as (ii) installing 5G coverage in all areas. However, as we showed, the proposed solution is designed to work also with common 3G/4G enabled phones and network types, (e.g., 3G and 4G only available in rural areas of the city), still within acceptable total latency of 760.94ms (3G) and 127.13ms (4G) on average for QoS 1. Furthermore, the cost of edge hardware integration should be in the range of € 200 on average for edge configuration such as RPi 4B, 8MP Camera Module V2 and Edge TPU.

5 RELATED WORK

The work [5] addressed multi-object tracking in urban settings using a network of edge devices. However, the discussed use case only considered privacy, accuracy and performance. In [11], the authors propose EdgeEye, a service enabling the development and execution of video analytics applications. EdgeBox [12] is an architecture for improving automatic event detection in edge near real-time video analytics. Authors in [3] discussed video image processing algorithms for real-time tracking and counting vehicles using edge devices. However, these works either do not consider strict low-latency requirements, or have analytics placement using cloud, or target different problems than traffic safety.

Vehicular networks and Vehicle-To-Everything (V2X) communication capabilities based on 5G are investigated in [14]. It introduces a novel system design, targeting Vehicles and roadside infrastructure with V2X capabilities. To increase traffic safety, the concept of [17] looked at the problem from a perspective of intelligent driving vehicles and real-time lane-change recognition. We focus on real-time edge analytics and high transmission mobile networks in detecting critical situations of pedestrians and cyclists appearing in drivers' blind spots on intersections.

6 CONCLUSIONS AND FUTURE WORK

We show the potential of edge computing and 5G to increase traffic safety. We target critical situations on intersections, e.g., pedestrians and cyclists appearing in drivers' blind spots. After analyzing design choices, we developed and evaluated a real-time edge analytics prototype. In the real-world setup, critical situations are detected and timely forwarded to mobile devices of drivers approaching the intersection. The proposed system enables real-time (i) detection of critical situations by running object detection on lightweight edge nodes; (ii) delivery of resulted critical information to vehicle drivers in 108.73ms on average with 5G. Further, InTraSafEd5G is designed to preserve privacy and guarantee low latency with other network types, representing a promising step for future edge applications and advances of communication technologies to support real-time decision-making.

The main limitations of this work are (i) the driver's app automatically subscribes to topics of certain intersections and (ii) the real-time notifications start showing when drivers' distance is 100m from the critical intersection. In future work, we plan to investigate location-based subscriptions [8] and the effect of distance and warning timings on drivers' brake reaction time, as in [16]. Furthermore, we plan to improve the resilience of critical edge processing to network/node failures (e.g., investigating container-based approaches for edge analytics).

ACKNOWLEDGMENTS

The work described in this paper has been partially funded through the Rucon project (Runtime Control in Multi Clouds), FWF Y 904 START-Programm 2015, 5G Use Case Challenge InTraSafEd 5G (Increasing Traffic Safety with Edge and 5G) funded by the City of Vienna and supported through Ivan Lujic's netidee scholarship by the Internet Foundation Austria.

REFERENCES

[1] Dovilé Adminaité-Fodor and Graziella Jost. 2020. *How safe is walking and cycling in Europe? PIN Flash Report 38.* European Transport Safety Council (ETSC).
[2] Wael KM Alhajyaseen and Miho Iryo-Asano. 2017. Studying critical pedestrian behavioral changes for the safety assessment at signalized crosswalks. *Safety science* 91 (2017), 351–360.
[3] Mallikarjun Anandhalli and Vishwanath P Baligar. 2018. A novel approach in real-time vehicle detection and tracking using Raspberry Pi. *Alexandria engineering journal* 57, 3 (2018), 1597–1607.
[4] Jiasi Chen and Xukan Ran. 2019. Deep Learning With Edge Computing: A Review. *Proc. IEEE* 107, 8 (2019), 1655–1674.
[5] Matthew Danish, Justas Brazauskas, Rob Bricheno, Ian Lewis, and Richard Mortier. 2020. DeepDish: Multi-Object Tracking with an off-the-Shelf Raspberry Pi. In *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking (EdgeSys '20).* ACM, 37–42.
[6] Krzysztof Goniewicz, Mariusz Goniewicz, Witold Pawłowski, and Piotr Fiedor. 2016. Road accident rates: strategies and programmes for improving road traffic safety. *European journal of trauma and emergency surgery* 42, 4 (2016), 433–438.
[7] Marc Green. 2000. " How long does it take to stop?" Methodological analysis of driver perception-brake times. *Transportation human factors* 2, 3 (2000), 195–216.
[8] Jonathan Hasenburg and David Bermbach. 2020. GeoBroker: A pub/sub broker considering geo-context information. *Software Impacts* 6 (2020), 100029.
[9] Roger A Light. 2017. Mosquitto: server and client implementation of the MQTT protocol. *Journal of Open Source Software* 2, 13 (2017), 265.
[10] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. *CoRR* abs/1405.0312 (2014). arXiv:1405.0312 http://arxiv.org/abs/1405.0312
[11] Peng Liu, Bozhao Qi, and Suman Banerjee. 2018. Edgeeye: An edge service framework for real-time intelligent video analytics. In *Proceedings of the 1st international workshop on edge systems, analytics and networking.* ACM, 1–6.
[12] Bing Luo, Sheng Tan, Zhifeng Yu, and Weisong Shi. 2018. EdgeBox: Live edge video analytics for near real-time event detection. In *2018 IEEE/ACM Symposium on Edge Computing (SEC).* IEEE, IEEE, 347–348.
[13] Philipp Schulz, Maximilian Matthe, Henrik Klessig, Meryem Simsek, Gerhard Fettweis, Junaid Ansari, Shehzad Ali Ashraf, Bjoern Almeroth, Jens Voigt, Ines Riedel, et al. 2017. Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture. *IEEE Communications Magazine* 55, 2 (2017), 70–78.
[14] Zunyi Shang. 2021. Low Latency V2X Application of MEC Architecture in Traffic Safety. In *The 2020 International Conference on Machine Learning and Big Data Analytics for IoT Security and Privacy*, John MacIntyre, Jinghua Zhao, and Xiaomeng Ma (Eds.). Springer International Publishing, 735–739.
[15] SWARCO. 2018. SIGNALS FOR THE FUTURE. LED traffic lights from the global market leader. https://www.swarco.com/sites/default/files/public/downloads/2019-02/SWARCO_folder_signalgeber_EN_screen.pdf (accessed: 2021-02-07).
[16] Xuedong Yan, Yuting Zhang, and Lu Ma. 2015. The influence of in-vehicle speech warning timing on drivers' collision avoidance performance at signalized intersections. *Transportation research part C: emerging technologies* 51 (2015), 231–242.
[17] K. Yu, L. Lin, M. Alazab, L. Tan, and B. Gu. 2020. Deep Learning-Based Traffic Safety Solution for a Mixture of Autonomous and Manual Vehicles in a 5G-Enabled Intelligent Transportation System. *IEEE Transactions on Intelligent Transportation Systems* (2020), 1–11. https://doi.org/10.1109/TITS.2020.3042504