Evaluation of AI-based Smart-Sensor Deployment at the Extreme Edge of a Software-Defined Network

Anton Aguilar, Ricard Vilalta, Raúl Parada, Fermin Mira
Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA)
Castelldefels, Spain

[aaguilar,ricard.vilalta,raul.parada,fermin.mira]@cttc.es

Francisco Vázquez-Gallego
i2CAT Foundation
Barcelona, Spain
francisco.vazquez@i2cat.net

***Abstract*—The introduction of AI-based smart-sensors on the network might suppose stringent requirements for the network edge, including the necessity to process real-time video feeds. Moreover, the introduction of vehicular communications allows the multiple location placement of necessary computational processes. To this end, we have proposed an AI-based smart-sensor solution that is able to be deployed at the extreme edge of the network (i.e., on the vehicle). The architecture for the connected vehicle is presented and accuracy results are provided for the proposed smart-sensor.**

***Index Terms*—Connected vehicles, V2X, Machine-learning, Hazard detection.**

I. INTRODUCTION

The introduction of vehicle to everything (V2X) communications has facilitated the spread of connected vehicles. This connectivity can be understood as a new safety mechanism that enhances the perception environment at a larger extend from the immediate surroundings of the vehicle [1]. With the advance of this technology, novel services that consider vehicle positions and detected hazards have appeared, such as Anticipated cooperative collision avoidance (ACCA), which considers a service architecture with two connected vehicles from original equipment manufacturers (OEMs) [2]. Some of the services to provide information require a large amount of computing capacity, such as computer-based vision for hazard detection. These computing capacity might be located at the network edge, as presented by the authors in [3]. The introduction of processing capabilities at network edge, might not be sufficient due to the necessary stringent network constraints that might require a continuous feed of stereo cameras through the V2X network. To this end, this paper proposes to introduce smart-sensors that are able to run at the extreme edge of the network and evaluate their accuracy.

This paper presents an architecture for detection of hazard events for a safer vehicle environment and their notification to other connected vehicles using V2X communications. The authors propose the introduction of Graphics processing unit (GPU) processing at the extreme edge of the network (i.e., the vehicle) to process video feed from a stereo-vision depth-camera in the front of it. For the experiments, the authors have used the GPS coordinates provided by a commercial device to have a reference to compare the accuracy of the proposed approach. It was concluded the proposed approach is suitable to use in highway scenarios.

This paper is organized as follows: In Section II we present a State-of-the-Art review. Section III describes the proposed architecture. Section IV provides the experimental evaluation, and, finally, Section V concludes this work.

II. STATE-OF-THE-ART

The detection of hazard events for a safer vehicle environment has been explored in the literature. Oh et al. [4] propose a rear-end collision risk index from data collected using a surveillance system in a freeway in California. This surveillance system uses multiple inductive loops placed on the pavement and video camera devices to obtain individual vehicle information such as model, vehicle arrival time, speed,etc. This system requires installation of inductive loops in the pavement, which could only be done for limited segments of the road.

A different approach is provided by Ahmed et al. [5] who used on-board active tags to track vehicles within toll roads. Again, collision estimation could only be computed in specific road segments where tag readers were installed. Sun and Sun [6] proposed a dynamic Bayesian network (DBN) model to predict crash events from video surveillance records in expressways.

Using a simulated in-vehicle system composed of radar, LiDAR and cameras, Dávid et al. [7] introduce a collision risk scale. The risk of basic driving actions such as overtaking is evaluated using simulations. Katrakazas et al. [8] perform risk evaluation using real-driving data obtained from a car equipped with GNSS and radar. The vehicle's speed its geo-spatial coordinates while the radar estimates the number of surrounding vehicles. Unfortunately, radar cannot identify and track the vehicles independently. Our Smart Sensor system covers the above disadvantages by identifying, classifying and tracking multiple vehicles simultaneously.

III. CONNECTED VEHICLE ARCHITECTURE

In this work, we propose using a machine-learning approach for hazard detection in real-time, leveraged by V2X technology to provide warnings with lane-accuracy. The system will be equipped on-board a connected vehicle. Warnings could be used to inform the driver, via an Human-Machine Interface (HMI), other connected vehicles nearby, and road operators. These warnings could become the base of other

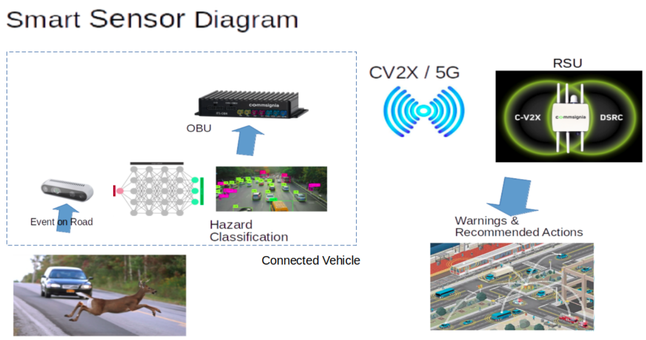

Fig. 1. Description of V2X leveraged hazard detection sensor.

V2X services, like hazard-notification, coordinated maneuver recommendation, platooning, and others.

A general description is shown in figure 1. Hazard events are detected using a depth-camera, which provides both RGB video and points-clouds of the surroundings. The latter will be used to estimate the location and speed of events relative to the vehicle. These data becomes the input of a machine-learning based algorithm (described below) to perform object classification, position estimation, and hazard detection in real-time. The algorithm also takes inputs from an on-board unit (OBU), which will provide data about the location, speed, and heading of the vehicle to compute the latitude and longitude of hazard's location. Besides, the OBU could be used to broadcast warning messages, which could be used by others to take preventive action.

1) On-board Hazard Detection: A detailed description of the on-board hazard detection system (a.k.a Smart-sensor) appears in figure 2. Inputs come from the depth-camera and the OBU. The former provides RGB video which is used for the neural network to perform classification of common objects in highway scenarios (e.g. vehicles, pedestrians, livestock). The classification algorithm is a neural network based on tiny YOLOv4 [9].

Our approach to tracking is defined from some assumptions about the nature of hazards: Hazards are considered punctual events. This occurs because the vehicle is assumed to be moving constantly forward, and any event will be relevant for the vehicle during a small time window depending on the vehicle-event relative speeds. Also, the camera is assumed being facing in the same direction of the vehicle's movement.

Points clouds from the depth-camera are used to estimate the relative position of events relative to the vehicle. These data are processed by the object tracker block which is a set of Kalman filters. These filters provide estimation of position and speed of objects detected by YOLO. Once the distance is estimated, we can use this information to obtain the perpendicular distance of the event relative to the vehicle using the points cloud, and use this data to compute the coordinates of the event.

Tracking is performed by the filters bank block, which decides when to create or delete Kalman filters. The process is not as rigorous as the one used in other tracking algorithms (i.e., Deep Sort [10]) to avoid overhead, decreasing frame-rate. Low frame-rates could hinder the sensor detecting and tracking objects at when the connected vehicle is moving at high speeds.

In our approach, assignation of bounding boxes from YOLO to Kalman filters is performed using the IoU metric [11]:

$$\mathrm{IoU} = \mathrm{argmax}_{i \in L} \frac{\left|B \cap B^i\right|}{\left|B \cup B^i\right|}, \tag{1}$$

where B are coordinates of current bounding-box associated to a Kalman filter, and B^i are the coordinates of the i-th candidate bounding-box. Equation 1 provides a reliable tracking device under the assumption the frame-rate is high enough, keeping uncertainty from Kalman filters under safe limits. Attaining higher frame-rates is desired to be able tracking faster moving objects. Under the conditions described, Kalman filters could be deleted once the hazard event has disappeared from view or they are not updated for times up to 1s. This value was empirically chosen based on tests results collected during the development.

The hazard warning generator block decides whether the tracked objects should be labeled like hazards and reported to the user. In our case, we use the class of the detected object, where classes "pedestrian" and "animal" are automatically considered hazards, in the case of vehicles (e.g., "car", "truck", etc) are considered hazards where they are stationary. The vehicle's speed is necessary in this step, which is provided by the OBU's sensors (i.e., accelerometers and GNSS). The "pedestrian" and "animal" classes are considered hazards automatically because they are not supposed to be present on the road; it should be noticed the area of interest in the image is the part focused on the highway.

Finally, detected hazards are reported using their coordinates, which are computed from the current vehicle's position and the filtered measurements from the depth-camera. We can compute the hazard's latitude and longitude difference from the vehicle's current position using the following equations [12]:

$$\begin{bmatrix} \Delta\mathrm{Lo} \\ \Delta\mathrm{La} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} \tag{2}$$

where equations 2 is basically a rotation matrix. z is the parallel component of the distance between the event and the camera, while x is the orthogonal component. θ is the vehicle's heading. Once we have the components the new coordinates are computed in the following manner:

$$\mathrm{La} = \mathrm{La}_0 + \frac{180\Delta\mathrm{La}}{\pi r_e}, \tag{3}$$

$$\mathrm{Lo} = \mathrm{Lo}_0 + \frac{180\Delta\mathrm{Lo}}{\pi r_e \cos\left(\frac{\pi\Delta La}{180}\right)}. \tag{4}$$

Where La_0 and Lo_0 are the vehicle's current coordinates, and r_e is Earth's radius in meters.

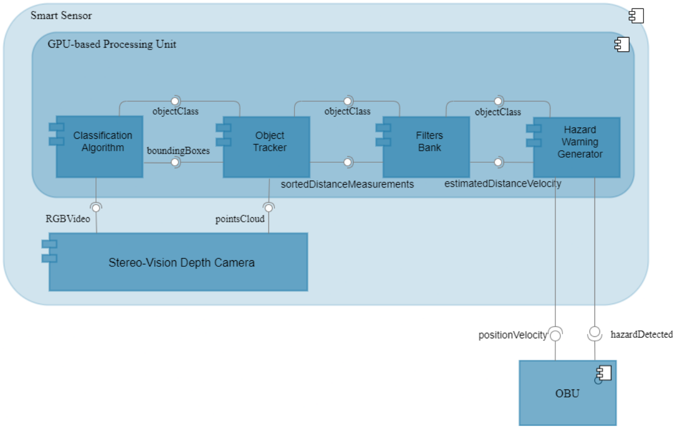

Fig. 2. Block Diagram of Smart-sensor.

2) On-Board Unit (OBU): The OBU has two main functions into the connected vehicle's architecture: Providing data to the Smart sensor and providing connectivity to send/receive V2X messages. The first point was explained above. Basically, the OBU provides data about vehicle's position, speed, and heading that is used to process the data from the depth-camera. On the other hand, the OBUs provide V2X connectivity that enables the vehicle to communicate with other connected vehicles or with the infrastructure using standard ETSI messages [13]. This is expected to become the base of Intelligent Transportation (ITS) Systems services in the near future.

In our application, the system relies on the Cooperative Awareness Message (CAM) and Decentralized Environmental Message (DENM) to run the application. While the information in these messages could be used to determine whether other connected vehicles could be a hazard, the connected vehicle should rely on the Smart-sensor when dealing with non-connected agents in the road (e.g. pedestrians, animal, legacy vehicles, etc). Our system is constantly monitoring CAM messages to further inform the user about their behavior. Moreover, DENM messages are used to send/receive warnings about hazards on the road. Each time a hazard is confirmed by the Smart sensor, a DENM is generated with the timestamp, location, and cause code of the event. In the meanwhile, the OBU is constantly scanning DENM messages sent from other connected vehicles and trigger warning messages to the user. The service could be easily extended using other messages as well, for example, the vehicle could receive MAPEM messages to further process hazard's location to indicate the user on which lane the event took place.

3) Dashboard: For the representation of the data we make use of the framework Streamlit [14] that allows displaying data in a website. It is Python-based and open-source. Once the application is launched, it can be accessed by a browser from different devices connected to the vehicle's network. Internet connection is needed to load the maps from the road.

In Figure 3 we can see a screenshot of the dashboard, the map is provided by OpenStreeMaps. At the top of the dashboard, we show the total number of connected vehicles and information about the last received warning and recommended action. On the map, we represent the vehicle with a blue arrow, and other vehicles with red arrows. Map is centered on the vehicle. Also, warning icons are represented temporarily on the map, with one hazard for animal and another hazard for slow vehicle in this particular example. By placing the mouse over icons or vehicles, we can see the identifier of other vehicle or the warning sender, and their distance to the vehicle. Additional information can be added in form of tables below the map.

Fig. 3. Displayed information in the dashboard.

4) MQTT Network: The system entities share the information from each other through a Message Queuing Telemetry Transport (MQTT) architecture. This is a widespread and lightweight standard messaging protocol designed around the concept of publishing/subscribing to topics and a broker for handling messages with connected IoT devices. The network is set using a WiFi hot-spot provided by the OBU, and the messages are encoded using JSON format. Figure 4 shows the MQTT messages exchanged between the entities described above.

Below, we provide a description of the MQTT messages shown in Figure 4. Each message is represented in JSON format including their data fields and a brief description:

- ***hazardDetected*** (from Smart Sensor to OBU):
  - *ObjectClass* Class of object detected by the Smart Sensor (e.g., vehicle, pedestrian, traffic sign).
  - *ObjectCoordinates* Estimated geographical coordinates of the object detected (latitude and longitude

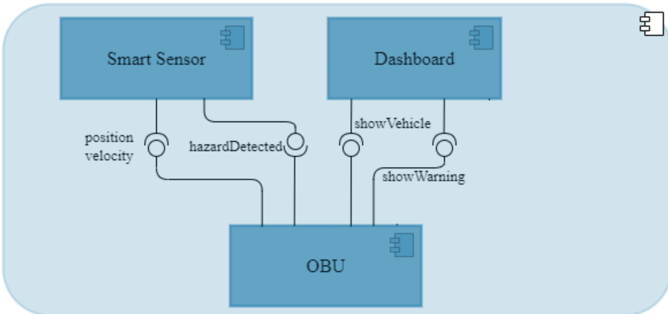

Fig. 4. MQTT messages'exchange

in degrees).

- *ObjectVelocity* Detected object's speed in the direction of the vehicle (km/h).
- *HazardType* If the object is considered a hazard, it indicates its type (stationary vehicle, slow vehicle, pedestrian).
- *DetectionTime* Timestamp when the object was detected (milliseconds).

- ***positionVelocity*** (from OBU to Smart Sensor):
  - *EgoVehicleCoordinates* Geographical coordinates of the vehicle (latitude and longitude in degrees).
  - *EgoVehicleVelocity* Speed of the vehicle (km/h).
- ***showWarning*** (from OBU to Dashboard):
  - *HazardType* Type of hazard detected on the road (stationary vehicle, slow vehicle, pedestrian, animal).
  - *HazardCoordinates* Estimated geographical coordinates of hazard (latitude and longitude in degrees).
  - *DetectionTime* Timestamp when hazard was detected (milliseconds).
- ***showVehicle*** (from OBU to Dashboard):
  - *VehicleCoordinates* Geographical coordinates of another connected vehicle (latitude and longitude in degrees).
  - *VehicleVelocity* Speed of another connected vehicle (km/h).
  - *VehicleHeading* Heading of another connected vehicle (in degrees).
  - *VehicleType* Type of another vehicle (e.g., car, bus, truck, etc).

IV. Experimental evaluation

The complete system was tested on the road to investigate its performance. Two vehicles were equipped with a commercial OBU (Commsginia OB4 [15]), and the smart sensor was equipped in one of them. It was implemented using an Intel Realsense D455 depth-camera and a Nvidia's Jetson Nano to run the machine learning software. The OBU model was chosen because the manufacture provides a commercial hardware with a complete protocol stack API that provides full freedom for development. On the other hand, the depth camera and GPU model were chosen having low power consumption in mind. The MQTT server and the dashboard were running on a laptop. The target vehicle was kept stationary while the camera vehicle moved and took measurements about its location from different positions and at different speeds (between 30 to 40 Km/h given the testing place is a parking lot). The target vehicle's OBU was used to obtain its position, deemed it as ground truth. In a second version of the experiment, a pedestrian, whose position was determined using a mobile phone application, was used as target. In the latter case, the vehicle was stationary while the pedestrian moved around the scene. Figures 5 and 6 show examples of these measurements, although, the data of the different tests are aggregated in the results shown below. In both cases, the positions where hazards were detected for the first time are shown in the figure. Tracking is performed to avoid multiple detection of the same event only.

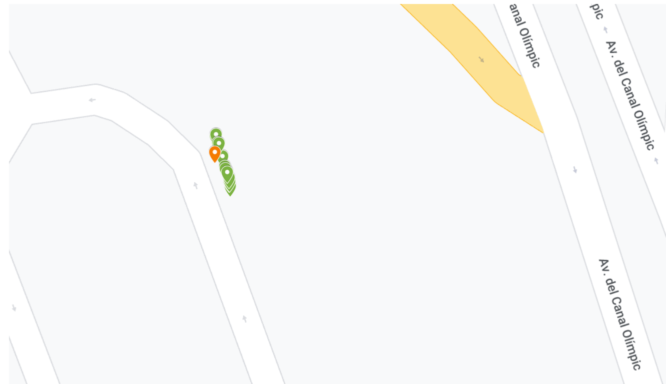

Fig. 5. Some samples from the target vehicle experiment (orange landmark is ground truth).

Target Pedestrian	Mean (m)	Standard Deviation (m)
Latitude	-0.689	1.486
Longitude	-0.588	1.322
Total	1.677	1.382
Target Vehicle	Mean (m)	Standard Deviation (m)
Latitude	-1.786	1.416
Longitude	1.474	0.541
Total	2.512	1.154

TABLE I
MEASUREMENT DEVIATIONS STATISTICS FOR TARGET VEHICLE AND TARGET PEDESTRIAN EXPERIMENTS ($1° \approx 111.374$KM).

Results are shown in table I. It shows the mean and standard deviation of the differences in latitude and longitude from ground truth for both experiments. The results are expressed in meters. The rows labeled "Total" show the mean and standard deviation for the Euclidean distance between the measurement and ground-truth points. These results show our approach is capable of hazard detection with lane-accuracy, assuming a lane has an approximate width of 3.5m in Spain [16]. The sensor operated at speed $\approx$ 19fps, at which level the assumptions made above seems to hold true.

We can see the results show less variation for the target pedestrian experiment than for the target truck experiment. This could be explained because in the truck experiment the camera vehicle is moving, while in the pedestrian experiment

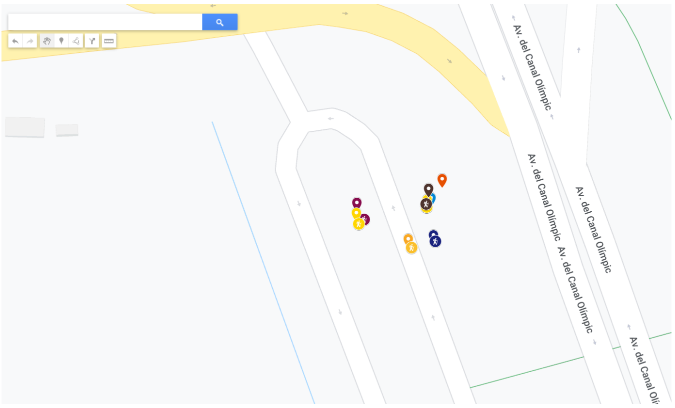

Fig. 6. Some samples from the pedestrian vehicle experiment (landmarks are ground truth).

it remained stationary. Basically, the uncertainty in the GPS measurements from the OBU is higher when the vehicle is moving, which is to be expected. In the end, uncertainty of the Smart sensor could not be less than the uncertainty of the GPS measurements used to compute the coordinates. Reporting the relative distances between the hazard and the vehicle in meters would lead to more precise results (no precision would be lost in conversions to latitude and longitude), but we feel these raw data would not be useful in highway applications, specially when we intend sending warnings to other connected vehicles.

Another point about the differences in accuracy comes from the nature of the Smart sensor itself. The Smart sensor performs classification of objects in the frame adding a bounding box around them, and the center of these boxes are used to obtain the corresponding depth point in the cloud to perform its computation. Therefore, depending on the particular position of the bounding box and the position of the vehicle there could be a variation on the measurement. For example, the measurement is expected to be different if the camera is viewing the rear part of the vehicle or the side part of it. On the other hand, a pedestrian is much smaller than a vehicle, and these effects are minimized. Therefore, higher precision at detecting pedestrians could be expected.

V. Conclusions

This paper introduced a novel architecture for hazard warning on connected vehicles using V2X communications. The authors proposed a GPU-based approach at the extreme edge of the network to process video feed of stereo-vision depth camera in the front of the vehicle. The results proved the feasibility and accurateness of the proposed solution. For the future work, the architecture could be extended to include a road side unit (RSU) to test communications performance. Frame-rate would be a critical factor, given low frame-rate will be an overhead to the communication to the infrastructure. Also, the hazard evaluation module could be enhanced with a more sophisticated methods like, for example, machine-learning algorithms.

Acknowledgment

This work is partially funded by the EC through the EC/5GPPP 5GMED (951947) project, the ”Grant RTI2018-099178-B-I00 funded by MCIN/AEI/ 10.13039/501100011033 and by ERDF A way of making Europe”, by the grant ONOFRE-3 PID2020-112675RB-C43 funded by MCIN/AEI/10.13039/501100011033, the Spanish UNICO-5G programm (TSI-063000-2021-41), and the regional FEDER FEMIoT P1(IU16-011655) project.

References

[1] M. Fallgren, M. Dillinger, J. Alonso-Zarate, M. Boban, T. Abbas, K. Manolakis, T. Mahmoodi, T. Svensson, A. Laya, and R. Vilalta, “Fifth-generation technologies for the connected car: Capable systems for vehicle-to-anything communications,” *IEEE vehicular technology magazine*, vol. 13, no. 3, pp. 28–38, 2018.

[2] F. Vázquez-Gallego, R. Vilalta, A. García, F. Mira, S. Vía, R. Muñoz, J. Alonso-Zarate, and M. Catalan-Cid, “A mobile edge computing-based collision avoidance system for future vehicular networks,” in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019, pp. 904–905.

[3] R. Muñoz, R. Vilalta, N. Yoshikane, R. Casellas, R. Martínez, T. Tsuritani, and I. Morita, “Integration of iot, transport sdn, and edge/cloud computing for dynamic distribution of iot analytics and efficient use of network resources,” *Journal of Lightwave Technology*, vol. 36, no. 7, pp. 1420–1428, 2018.

[4] C. Oh, S. Park, and S. G. Ritchie, “A method for identifying rear-end collision risks using inductive loop detectors,” *Accident Analysis & Prevention*, vol. 38, no. 2, pp. 295–301, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0001457505001570

[5] M. M. Ahmed, M. Abdel-Aty, and R. Yu, “Bayesian updating approach for real-time safety evaluation with automatic vehicle identification data,” *Transportation Research Record*, vol. 2280, no. 1, pp. 60–67, 2012. [Online]. Available: https://doi.org/10.3141/2280-07

[6] J. Sun and J. Sun, “A dynamic bayesian network model for real-time crash prediction using traffic speed conditions data,” *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 176–186, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X15000856

[7] B. Dávid, G. Láncz, and G. Hunyady, “Real-time behaviour planning and highway situation analysis concept with scenario classification and risk estimation for autonomous vehicles,” *Designs*, vol. 3, no. 1, 2019. [Online]. Available: https://www.mdpi.com/2411-9660/3/1/4

[8] C. Katrakazas, M. Quddus, and W.-H. Chen, “A new integrated collision risk assessment methodology for autonomous vehicles,” *Accident Analysis & Prevention*, vol. 127, pp. 61–79, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0001457518306614

[9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020.

[10] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[11] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized intersection over union: A metric and a loss for bounding box regression,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 658–666.

[12] G. P. Gerdan and R. E. Deakin, “Transforming cartesian coordinates x, y, z to geographical coordinates ϕ, λ, h,” *Australian surveyor*, vol. 44, no. 1, pp. 55–63, 1999.

[13] T. ETSI, “Intelligent transport systems (its); vehicular communications; basic set of applications; part 2: Specification of cooperative awareness basic service,” *Draft ETSI TS*, vol. 20, no. 2011, pp. 448–51, 2011.

[14] [Online]. Available: https://streamlit.io/

[15] C. Ltd. (2022) Powerful v2x onboard unit. [Online]. Available: https://www.commsignia.com/products/obu/

[16] J. HINOJOSA CABRERA, “La nueva norma 3.1-ic” trazado” de la instruccion de carreteras,” *CARRETERAS, REVISTA TECNICA DE LA ASOCIACION ESPANOLA DE LA CARRETERA*, no. 94, 1998.