

Integrating and Evaluating the TruePAL AI Assistant for First Responders

Christian G. Hahm

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA, USA
Temple University
christian.hahm@temple.edu

Abstract. TruePAL (Trusted and Explainable AI for Saving Lives) is a research project exploring how various modern AI methods can be used to assist first responders (e.g. police officers, firefighters, and EMS) in their vehicles and at the emergency scene. Importantly, TruePAL exhibits trustworthy behavior more so than many AI systems since it uses reasonable and human-understandable explanations for its actions and decisions. The project’s individual components have each undergone their own planning, development, and testing. For the final product, we integrate the multiple components into a single system that can be easily downloaded and executed on a first responder vehicle’s on-board computer and mobile phone. The integrated TruePAL system combines 1.) vehicle simulator (CARLA) sensor data (i.e. camera and RADAR data), 2.) a deep convolutional neural network (YOLOv4) for visual object detection, 3.) a Kalman filter for visual object tracking, 4.) an artificial general intelligence reasoning system (NARS), and 5.) I/O communication with a backend database and frontend mobile application. The integrated system is evaluated on a suite of test cases, which we established in collaboration with first responders and the U.S. Department of Transportation. We show that TruePAL is able to provide robust, timely, and explainable situational awareness alerts that may prevent collisions.

1 Introduction

First responders work in a high-risk and stressful field; they are the ones meant to quickly and effectively receive news of, travel to, and manage emergency situations. The journey to an emergency scene is a hectic one, since the response is often time-critical (e.g. due to a blazing fire, person in cardiac arrest, active shooter, etc.) and the first responder must drive their vehicle quickly enough to achieve a rapid response but slow enough to complete the travel safely. As a consequence of this dangerous but necessary haste, first responders are at a higher risk for vehicle collisions and fatalities than the average driver. Police officers, for example, are nearly twice as likely as the average person to be involved in a vehicle crash. Half of all law enforcement deaths are due to vehicle accidents. Firetruck collisions are the second leading cause of death for firefighters. Overall, a fatality caused by a vehicle collision is five times more likely to have occurred to a first responder than the average person. In addition to the tragic loss

of life, emergency vehicle accidents in the United States alone are estimated to cost \$35 billion annually.¹

The TruePAL research project seeks to ameliorate dangerous emergency response transits by combining multiple modern AI techniques into a single integrated system that acts as an autonomous assistant for first responders. The system runs on a vehicle's on-board computer, is internet-connected, and communicates with backend databases and the driver via heads-up display (HUD) and our custom mobile application. The primary function of TruePAL is to work alongside the first responder to provide automated situational awareness and communicate notifications that may help protect the first responder from unexpected or unnoticed dangerous driving situations. The system constantly monitors inputs from various data sources (especially on-board vehicle sensors like cameras and RADAR) while integrating and logically reasoning on the information in real-time.

2 Methods and Results

2.1 Integration of Individual Components

In order to produce a functional integrated system, each of the components must be separately developed, tested, and verified before they can be integrated and expected to work reliably. After the planning phase of TruePAL, the research group was split into various teams which completed their respective components. The individual components of the overall TruePAL application include:

- A **vehicle simulator** and **sensors** (CARLA driving simulator²) to provide raw sensory inputs (e.g. vision and RADAR) and a realistic driving environment.
- A **visual object detector** (deep convolutional neural network YOLOv4³) detecting vehicles, traffic lights, etc. in an RGB camera image.
- A **visual object tracker** (Kalman filter) for object persistence, tracking individual instances of objects, their IDs, and their movements using RADAR sensor data and bounding box detections. Together with the neural network, the two vision modules roughly act as a “visual cortex” that pre-processes visual information to be used at a later stage.
- Artificial general intelligence **reasoning system** (NARS⁴) which acts as a central explainable “prefrontal cortex”, making decisions to achieve its goals while integrating and reasoning about its perceived experience. This module receives inputs from the object tracker + sensors, and interacts with the first responder and backend database.
- A frontend **mobile application** with an AI chatbot, hazardous materials sign detection, information lookup, and notifications from the TruePAL desktop computer system.
- A backend **database** to which all TruePAL agents connect. TruePAL draws knowledge from it (such as the location of other nearby TruePAL agents, intersection / roadside collision history, etc.) and uploads to it (vehicle location for real-time map updates in the TruePAL app, audit trail of TruePAL's actions and explanations, etc.)

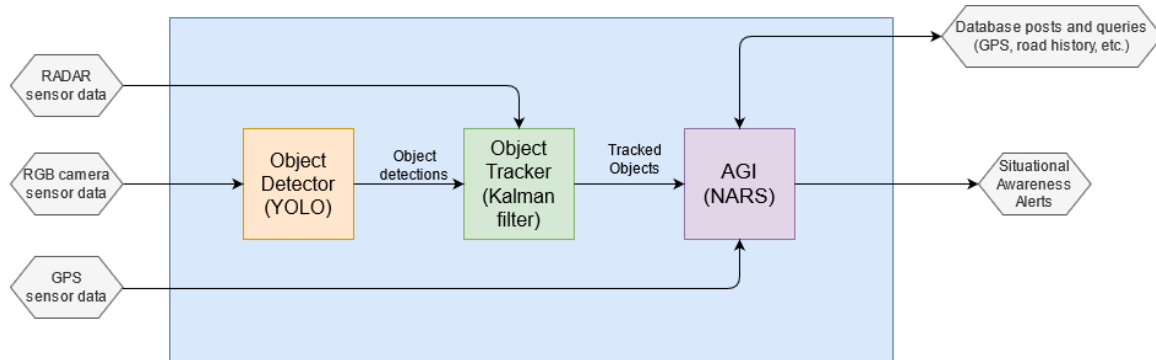


Fig. 1: Architecture of the TruePAL Desktop Application

Figure 1 provides a clear overview of the integrated system’s design. The team made the smart decision to use a common language, Python, for each component, which makes data flow between the components easy and efficient. The integrated TruePAL system takes advantage of Python’s multiprocessing capabilities for high computational performance, user friendliness, and to enforce clean modularity. The CARLA simulator, CARLA “manual control” window, and TruePAL system each run on their own separate process, but are all launched by the same program – since each is executed as a separate process, each is provided with its own set of dedicated resources by the operating system. Importantly, running each program on a separate *process* allows this functionality, as opposed to executing each program on a separate *thread* which, despite allowing multiple programs to run concurrently, does not allocate more resources to each program, meaning each of the threads must compete for their parent process’ finite resources and will not result in computational speedup.

The CARLA simulator computes everything required to simulate the driving environment, whereas the manual control program is how 1.) the user controls their vehicle and 2.) TruePAL accesses the vehicle’s sensor data. When executing the manual control process, TruePAL passes in multiprocessing queues (a special FIFO data structure that can transfer data between processes) as parameters. We modified the CARLA manual control program for our unique purposes in many ways, and use the multiprocessing queues in asynchronous sensor callbacks, where on each callback we “put()” the sensation onto the queue. Concurrently, the TruePAL program itself executes an infinite main loop during which it processes the sensations; in each iteration, the system updates every sensor queue (by performing “get()” until only one element remains), and processes the most up-to-date sensation received from each queue.

The TruePAL agent sends its GPS data to the server database at a fixed rate (once every second), and also itself saves the data for processing. When a RGB camera image is available from the camera sensor, the image is run through the visual object detector (YOLOv4) which finds, marks, and labels objects in the image. After this stage, TruePAL has the bounding boxes and *class* labels for relevant objects in the scene (e.g. other vehicles, traffic lights). However, this is not enough for scene understanding, since in a driving scenario it is necessary for TruePAL to additionally keep track of the in-

dividual object *instances* in the scene as well as track their movements and estimate their velocities. After the object detection step, the information is passed to the Object Tracker (a Kalman filter) which performs these exact duties. This is the stage in which we integrate the RADAR sensory information, by finding common detections between RADAR and image. TruePAL uses the RADAR to more accurately estimate object *depth* and *velocity*, and also estimate bounding boxes for other (moving) RADAR detections which might represent vehicles missed by YOLOv4.

Finally, the scene information is encoded into JSON format which is further converted into Narsese, the language of NARS. NARS is a general theoretical framework, but has multiple digital implementations, which are still experimental and under development. For TruePAL, we use a C-based version of NARS called OpenNARS for Applications,⁵ aka ONA. As the name suggests, ONA is well suited for reliable performance in event-driven and sensorimotor applications. The Narsese inputs provide the system with real-time updates of its environment. NARS is constantly integrating and reasoning on these inputs, allowing it to make decisions in combination with its existing background knowledge (provided by the TruePAL team). It is simple for humans to understand the reasons behind why the system makes its decisions; this explainability is free as NARS is a reasoning system. Each of the system’s inference results, including its actions, are explained by the premises from which the results are derived. The Narsese premises can be converted to human-understandable English using the Narsese-to-English translator provided with ONA. The system’s actions and explanations are stored to the TruePAL database in case the need for audit arises.

TruePAL communicates with a backend database for a few reasons: to post its own information for storage, to gain information about other TruePAL units, to access historical road collision data, and to communicate with the frontend mobile app. The system’s alerts appear on the desktop application itself (in red text) and on the user’s mobile app, provided both the TruePAL desktop application and the mobile application are accessed using the same username. The mobile application also provides text-to-speech capability, reading notifications out loud which allows the driver to keep their eyes on the road.

2.2 System Evaluation and Test Cases

After integration, we evaluated the system using the test cases developed by the CARLA team based on the highest priority use cases as determined by first responders we interviewed. The test cases, also called “scenarios”, have multiple variations to evaluate the system’s robustness. In total there are 19 test cases, 13 of which are CARLA test cases and 6 of which are mobile app test cases. Many of the CARLA test cases have hard-coded variations (including location, vehicle behavior, etc.), for a total of 25 CARLA test cases.

TruePAL monitors many data sources at once to check for potential threats. A few of the scenarios include:

- **Scenario 1 - Intersection Navigation examples:**
 - *Scenario 1.1:* TruePAL queries a traffic accident history database to identify and warn about high-risk intersections.



Fig. 2: Scenario variations include locations and randomized civilian vehicle models.

- *Scenario 1.2*: TruePAL warns the first responder if they are zooming through a red light at a high speed (> 35 MPH). They should move slowly to safely clear the intersection.
- *Scenario 1.3*: TruePAL uses GPS coordinates to warn when approaching an intersection at the same time as another first responder / TruePAL unit.
- *Scenario 1.4*: TruePAL warns when on a collision path with another vehicle.
- *Scenario 1.5*: (NULL Case) TruePAL generates no warning when driving through a safe green light intersection at any speed.
- *Scenario 1.6*: TruePAL alerts when a vehicle in front is not yielding to the first responder.
- **Scenario 2 - Traffic Stop examples:**
 - *Scenario 2.2*: TruePAL alerts when a vehicle from behind is veering off the road and on a collision course.
 - *Scenario 2.3*: TruePAL detects when a vehicle is weaving between lanes.

In addition to our hardcoded variations, each test case is implemented with elements of randomness. For example, we randomize the model of vehicles in the scene to test the robustness of our object detection (see *Figure 2*); the trained neural network should be able to detect many different types of cars. Some scenarios have randomized elements of movement trajectory to stress the limits of our object tracking and alerts. Furthermore, we created a user configuration file to allow the user to set scenario parameters, such as the user vehicle's speed, the collision vehicle's speed (in scenarios with collisions), and their TruePAL username and login information. The user may also change weather conditions (e.g. rainy, cloudy, sunny) on the fly using their keyboard.

In the face of a complex integrated system, we developed tools such as clickable icons to provide a user-friendly experience. The TruePAL desktop system pushes its alerts / updates to the mobile app (see *Figure 3*) and also shows the user a visualization



Fig. 3: **Scenario 1.3.1:** TruePAL warns that the user is approaching the next intersection at the same time as another first responder / TruePAL vehicle. The warning is given far enough in advance that the first responder can brake and prevent the collision.

of what it's sensing and detecting (see *Figure 3, 4*). The visualization includes RADAR hits, detected traffic lanes, and detected + tracked bounding boxes of vehicles and traffic lights. TruePAL's warnings are also displayed on this view in red text (see *Figure 4*), in addition to appearing on the mobile app.

The project is still ongoing. We are towards the end of the testing phase and still actively testing and modifying the system. Currently, the system passes 21/25 of the test cases consistently. The system correctly perceives and understands its environment well enough to provide relevant alerts at the proper moment to assist the first responder, while remaining silent (but still actively monitoring and learning) when the situation is safe.

This report is written at the conclusion of Phase I, which is meant to demonstrate the value of TruePAL and develop a working prototype for future iterations. The future directions for this project may be realized in Phase II of the project, which involves bringing the system into the real-world, and testing it directly with real sensor-equipped first-responder vehicles. This transition should not require many changes, except to the object detection neural model which will require re-training on real-world images of vehicles and traffic lights. We may also add more capabilities to the system, such as emergency scene updates (ETA, arrival notifications, emergency scene icons automatically added to the mobile map, etc.). Future iterations of TruePAL might wirelessly communicate directly with other TruePAL agents by vehicle-to-vehicle (v2v) communication in addition to being web-connected. Overall, Phase I of the project is a success and the system is geared for release as well as future updates.

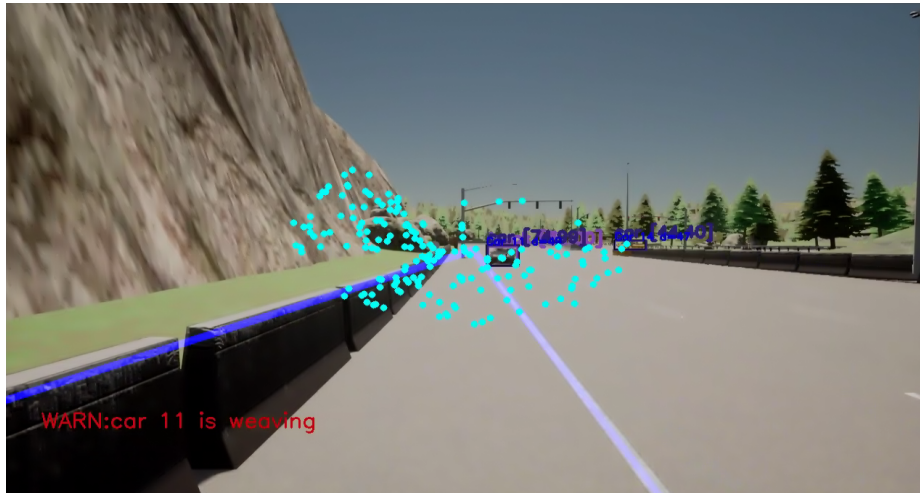


Fig. 4: **Scenario 2.3.1:** TruePAL warns about a dangerously weaving vehicle *Car 11* rear-ends the First Responder on the highway. (**Light Blue dots** = *RADAR hits*; **Red text** = *TruePAL warning*; **Blue lines** = *Detected lanes*; **Dark Blue Bounding box + label** = *Detected vehicle*)

3 Acknowledgments

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the JPL Summer Internship Program and the National Aeronautics and Space Administration (80NM0018D0004). I thank the TruePAL team and my JPL mentor Dr. Thomas Lu for his excellent management of the TruePAL project.

References

1. Fetto law group: Statistics on emergency vehicle accidents in the united states, 2018. <https://www.fettolawgroup.com/Personal-Injury-Blog/2018/March/Statistics-on-Emergency-Vehicle-Accidents-in-the.aspx>.
2. Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.
3. Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
4. Pei Wang. *Non-Axiomatic Logic: A Model of Intelligent Reasoning*. World Scientific, Singapore, 2013.
5. Patrick Hammer and Tony Lofthouse. ‘opennars for applications’: Architecture and control. In *International Conference on Artificial General Intelligence*, pages 193–204. Springer, 2020.