AV Car Testing Deliverable #1

CMPE 287 - 01 Spring 2022

Huyen Nguyen

Prit Lakhani

Richard Ngo

Sudha Vijayakumar

I. Introduction

Testing Concepts

Autonomous vehicles are the future of automation, and it is extremely important to ensure the safety of AV by testing and validating. The Society of Automotive Engineers International (SAE) had categorized vehicle automation into six levels, ranging from fully human driver hand-on to fully autonomous system without the need of driver's attention. The table 1 describes each SAE vehicle automation level.

Table 1: Vehicle Automation Levels

SAE Vehicle Automation Level	**Description**
Level 0 *No Driving Automation*	Human drivers are involved in everything. Vehicle is fully manual controlled.
Level 1 *Driver Assistance*	Level 1 is the lowest level of driving automation. There are some single automated system features in the vehicle such as steering or cruise control (accelerating). Human drivers are required to monitor non-automated aspects of driving.
Level 2 *Partial Driving Automation*	Level 2 means the vehicles provide advanced driver assistance for human drivers. Although both steering and accelerating/decelerating can be controlled by the vehicle, humans are required to be seated in the driver's seat and

	can take control of the car at any given time. Tesla Autopilot is currently at level 2 as well as Cadillac Super Cruise from General Motors.
Level 3 *Conditional Driving Automation*	Level 3 vehicles can detect the surroundings and make decisions without the driver's intervention. Human drivers have the ability to override the vehicle anytime. The system is not fully autonomous yet, so the driver must remain aware at all times and ready to take control when the system has technical issues. Audi A8 is currently the vehicle that has level 3 autonomous.
Level 4 *High Driving Autonomous*	Level 4 vehicles can function almost all the features smoothly without human interaction. The vehicles are intelligent enough to intervene if there is a technical issue in the system. Most of the time, human intervention is not required, but drivers can manually override the vehicles. Waymo is one of the vehicles that has achieved level 4.
Level 5 *Full Driving Automation*	Level 5 vehicles will be without dynamic driving tasks, which means no human attention is required. The system is fully autonomous with the ability to function with all the features that human can do.

In order to develop an autonomous vehicle, a lot of technologies are being involved including electronic sensors, employing 360-degree view cameras, using computer vision software etc. To successfully navigate roadways, vehicles need to achieve four tasks, which are to detect objects in the vehicle's surroundings, to identify those objects, to predict the objects' next moves, and lastly to plan an appropriate response in response to the objects.

One of the ways to test AV is using simulation testing. Simulation testing allows engineers to test in a wide range of scenarios. Engineers can modify the code after every scenario for better results and improve models without having to physically deploy real cars into the real world. There are different approaches of simulation testing include:

1. End-to-End Simulation: engineers have the ability to generate the entire AV synthetically.
2. Replay Simulation: can simulate real-world state. Engineers can modify the code to be more effective and behave well in the real world scenario.
3. Planning Simulation: AV system is not aware that it is in simulation and should know most things in the real world. This simulation allows engineers to observe how the model interacts with the assertion.

Testing Approaches

Scenario-Based Testing is a very popular approach in testing AV. LGSVL Simulator is an open-source code from LG Electronics America R&D Lab to provide a simulator for developers to test AV algorithms. LGSVL currently has integration with Autoware.auto and Apollo platforms. The combined integration system is able to generate maps, weathers etc. LGSVL Simulator allows engineers to build, test, and validate the model based on different scenarios. Some of the use case examples for using SVL Simulator include a test planning module in isolation with virtual ground truth detections, or keeping track of progress with test case reports

after each simulation. Section 3 will go into detail about each different scenario that we will be testing for this project.

Testing Challenges

There are a lot of challenges when it comes to testing AV, and the five major challenges are “drive out of the lanes, complex requirements, non-deterministic algorithms, inductive learning algorithms, and fail-operational systems” [1].

Beside software-related challenges, there are also physical and surrounding problems such as below:

- Road conditions - when people travel to different places, they usually don’t know the road condition beforehand. Some roads can have tons of potholes and some roads can be very nice. These unpredictable roads can be a problem since it can confuse the software if the model has not been trained for the road conditions.
- Weather conditions - just as road conditions, weather can vary from one place to another. Autonomous cars should work in all conditions, especially the harsh ones such as storms or raining.

Testing Requirements

In order to test LGSVL simulation, these are the requirements to test and validate.:

- Python
- At least 4 GHz Quad core CPU
- NVIDIA GTX 1080 (8GB memory) or higher
- Ubuntu 18.04 (64-bit) or Ubuntu 20.04 (64-bit)

II. SVL Simulator

Functions

SVL Simulator

SVL Simulator is a high-fidelity simulation environment for developing autonomous vehicles and robotic systems. SVL simulator aid end-end real-time simulation and helps developers interface real systems with the Simulator and conduct rigorous testing. The testing needs and infrastructure required to support safe systems on public roads and environments are becoming increasingly crucial as the development of sophisticated cars, transportation systems, and autonomous agents grow increasingly complicated. The simulation software, software tools, an ecosystem of content and plugins that enable tailored use cases, and the cloud architecture that enables simulation and scenario testing at scale are all part of the SVL Simulator. The simulation software provides a seamless and customized interface with a user's System Under Test by modeling a virtual environment, one or more ego cars or autonomous systems and their sensors, traffic, and other dynamic objects. This enables the developer to troubleshoot and execute other tasks. Engineers who design, test, and validate autonomous systems will benefit from the SVL Simulator. Integration testing, modular algorithm testing, and system verification are all possible using the SVL Simulator. It includes the testing infrastructure and methodology required to develop and deploy a safe autonomous vehicle or robot.

Applications

Some of the applications developed using the SVL simulator are as follows:

- Mobile robotics in the open air
- Services for Future Mobility
- Entirely autonomous racing
- Development and marketing of sensors and sensor systems
- L4/L5 self-driving car systems
- ADAS/AD systems L2/L3
- Robotics in the warehouse
- Security in autonomous vehicles and systems
- Generating synthetic data
- Automotive real-time embedded systems

Some of the practical use-cases of the SVL simulator are as follows:

- It helps to create and execute scenarios with complex traffic scenarios based on real-world data.
- It aids isolated test planning modules with virtual ground truth detections.
- It enables testing autonomous car stack in real-time from beginning to end (software-in-the-loop).
- It helps to find and avoid performance bottlenecks with hardware-in-the-loop simulation.
- It helps keep track of the development progress through test case reports for each simulation.

- It helps to find high-value and intriguing edge case scenarios, parameterize, and automate the execution of thousands of scenarios.

Simulation Options,

- Random Traffic Simulation
- API_Only Simulation
- Python API

Simulator Coordinate System

Transforms in Unity are used to express locations and rotations inside the Simulator. For transforms, Unity employs a left-handed ZXY coordinate system, with the Y-axis as the vertical axis. A transform will be connected with each simulation object, such as vehicles, sensors, maps, traffic signals, etc. It is also vital to understand the transformations when creating bespoke sensor combinations for ego vehicles.

Architectures

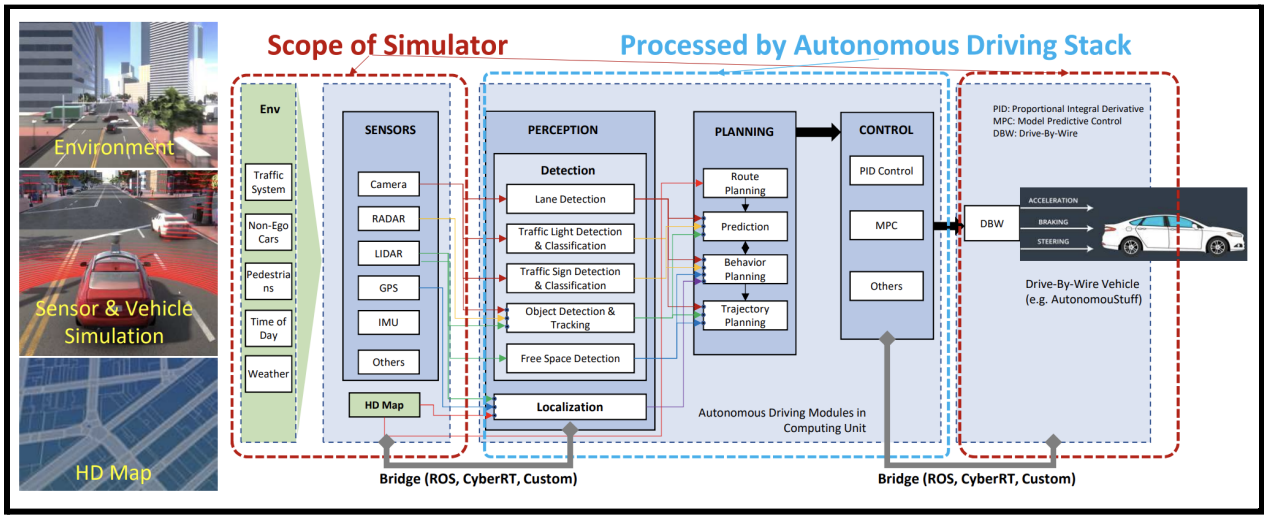


Image source:https://arxiv.org/pdf/2005.03778v1.pdf

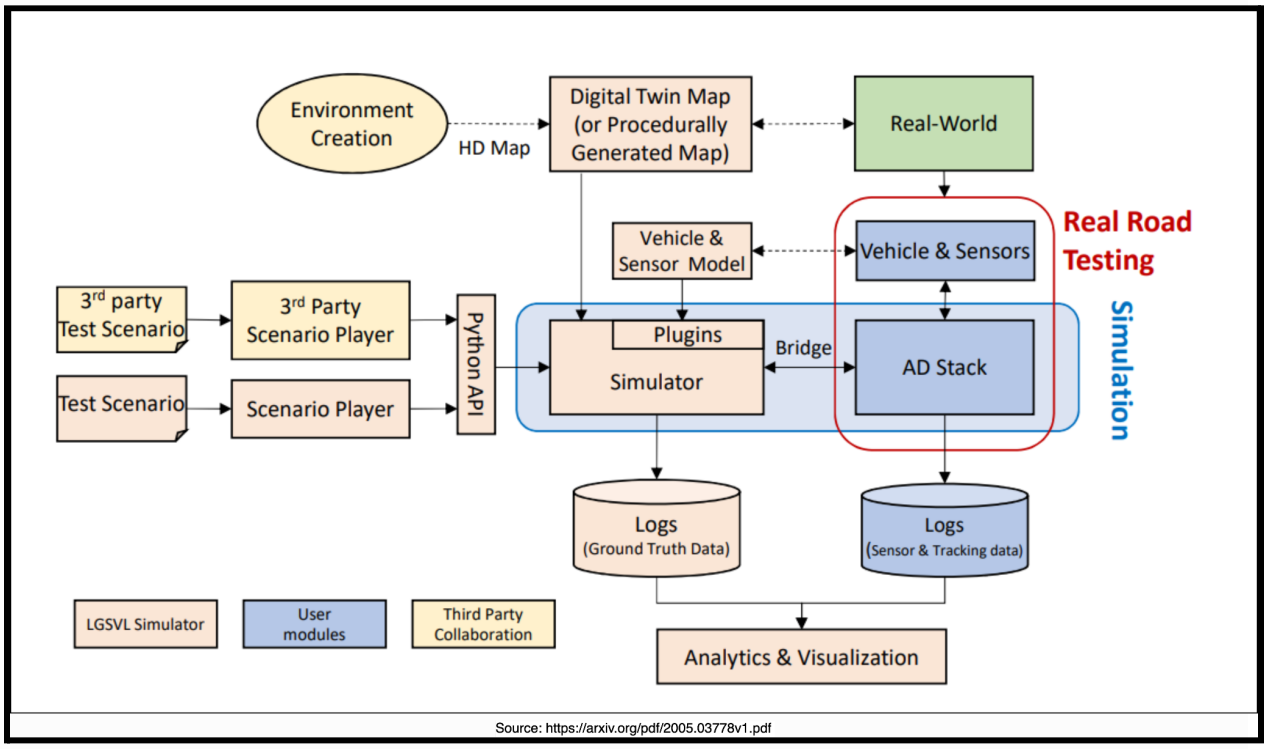
LGSVL Simulator is a simulation tool that uses Unity's game engine and takes advantage of the newest Unity technologies, such as the High Definition Render Pipeline (HDRP), to create photorealistic virtual environments that resemble the actual world. Environment simulation, sensor simulation, vehicle dynamics, and control simulation of an ego vehicle are all functions of the simulation engine. Traffic simulation and physical environment simulations such as weather and time of day are all part of the environment simulation. These elements are crucial in the simulation of test scenarios. The Python API may be used to control all elements of environment simulation. LGSVL Simulator's simulation engine is a free and open-source project. The LGSVL Simulator's ego vehicle sensor setup is entirely adjustable. Sensor settings are accepted as JSON structured text in the Simulator's online user interface, making it simple to set up sensors' intrinsic and extrinsic parameters. The sensor type, location, publication rate, topic name, and measurement reference frame are all described in each sensor entry. Some sensors may have extra fields to describe characteristics further; for example, the beam count of each LiDAR

sensor can be customized. The Simulator comes with a default set of sensors, including a camera, LiDAR, Radar, GPS, IMU, and many virtual ground truth sensors. Users may also create their unique sensors and use them as sensor plugins in the Simulator. Some of the sensors in LGSVL seen in Simulator are: the left column depicts physical sensors such as a fish-eye camera sensor, LiDAR sensor, and Radar sensor, while the right column depicts virtual ground truth sensors such as a segmentation sensor, depth sensor, and 3D bounding box sensor. We integrate semantic segmentation with instance segmentation for the segmentation sensor. Users can choose which semantics get instance segmentation; objects with these semantics will have distinct segmentation colors, but examples of other objects will only have one segmentation color per semantic. For example, if the user only enabled instance segmentation for "vehicle" and "pedestrian," all buildings will have the same segmentation color, while all roads will have a different segmentation color. Each automobile and pedestrian will have a separate color segmentation. However, all cars will be the same hue (e.g., all blue) and indistinguishable from pedestrians (e.g., all reddish).

In addition to the default reference sensors, LGSVL Simulator now supports real-world sensor models used in autonomous vehicle systems. These sensor plugins, such as the Velodyne VLP-16 LiDAR, contain settings that match their real-world equivalents and operate as a genuine sensor, providing realistic data in the same format. Users may also design their sensor plugins to add more variants and even new types of sensors that the LGSVL Simulator does not support by default. The LGSVL Simulator provides a rudimentary vehicle dynamics model for the ego vehicle. Furthermore, the vehicle dynamics system is built up to allow integration of external third-party dynamics models via a Functional Mockup Interface (FMI), shared libraries, and a Functional Mockup Interface (FMI). Furthermore, separate IPC (Inter-Process Communication)

interfaces for co-simulation can be imported into the Simulator. Consequently, users may utilize LGSVL Simulator in conjunction with third-party vehicle dynamics simulation software to obtain the best of both worlds.

LGSVL Simulator-enabled autonomous driving simulation workflow



User Active Directory Stack is the simulation-based system that the user wishes to design, test, and validate. LGSVL Simulator presently supports reference out-of-the-box integration with 3 Simulator's open-source AD system platforms Apollo3. Stack for Active Directory Map of Digital Twins (or Procedurally Generated Map) Logs from the Real World (Ground Truth Data) Logs are a kind of wood (Sensor & Tracking data) Player Simulation Scenario Test Scenario Testing on the Road Sensors & Vehicle Visualization & Analytics Plugins for Vehicle and Sensor

Models User modules for environment creation Simulator for LGSVL Collaboration with a third party HD Map of the Bridge Player in a 3rd-Party Scenario 3rd-Party Scenario Python API Test Scenario. The user AD stack communicates with LGSVL Simulator via a communication bridge interface, which is chosen based on the runtime framework of the user AD stack. A specific bridge is supplied to the Simulator for Baidu's Apollo platform, which employs a proprietary runtime framework called Cyber RT. Autoware.AI and Autoware are two different types of Autoware. Through common open-source ROS and ROS2 bridges, Auto, which runs on ROS and ROS2, may connect to LGSVL Simulator. Autoware and Apollo operate on LGSVL Simulator. A custom communication bridge interface may be readily introduced as a plugin if the user's AD stack employs a custom runtime framework. LGSVL Simulator also allows connecting numerous AD systems.

System Requirements

- ~4 GHz, Quad-core (or better), 16+ GB of RAM
- Nvidia GTX-1070/1080 ("Pascal") – works with Apollo 5.0 or newer
- Nvidia RTX-2070/2080 ("Touring") – requires Apollo 5.5 or newer
- Nvidia RTX-3060+ ("Ampere") – requires Apollo "master" (aka Apollo 6.1+)
- 8+ GB of GPU memory (large maps and sharing with Apollo requires more memory)
- Number of machines for real-time testing
- Two is better, but one (8+ GB GPU) should be usable (with "modular testing")
- Windows 10 or preferably Linux (Ubuntu 18.04/20.04) – or one of each

Current Features

Key featuresEnd-end simulation

- Simulation of a photorealistic environment
- High-performance, real-time simulation
- Simulation of multiple vehicles
- Customizability and extensibility
- Choice of vehicles, sensors, custom objects, and maps.
- Several tools for creating scenarios are available
- Tools for creating procedural road networks and environments
- Tools for importing, exporting, and annotating HD maps

Library

The Library page within the SVL Test system client interface shows the set of maps and vehicles included in the account, accessible for use in recreations. In addition, an instructional exercise video on including a resource from the Store and uploading claim resources can be found here. The library is the central place for all the content and information related to a profile's SVL Test

system_account.

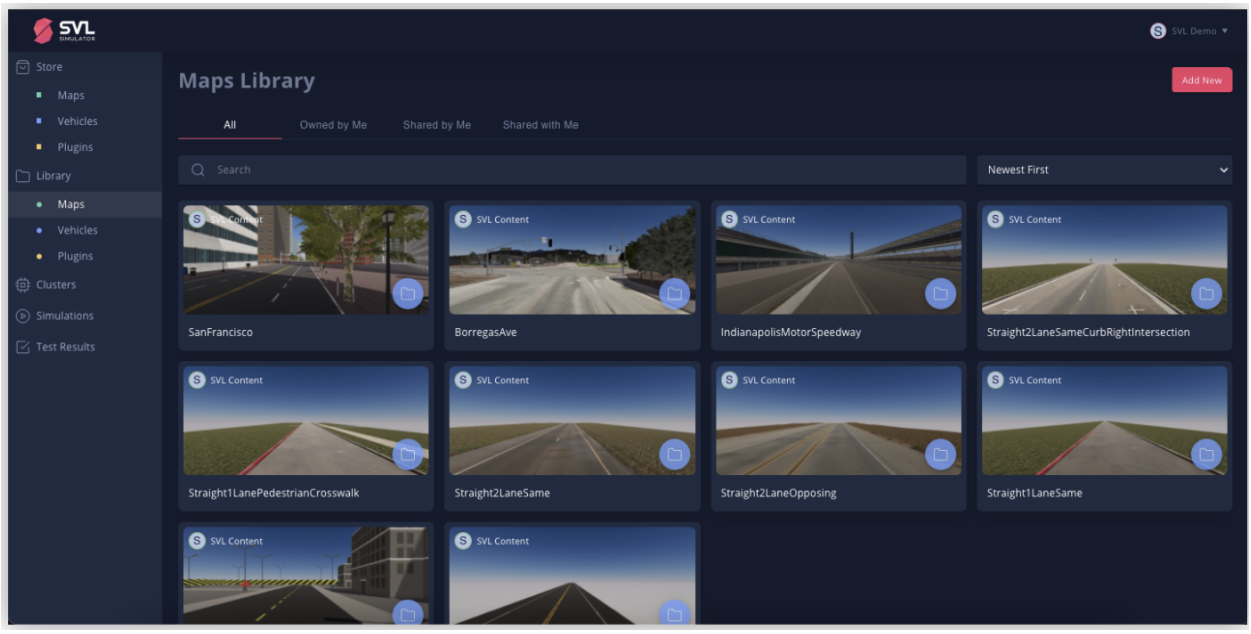


Image source: https://www.svlsimulator.com/docs/user-interface/web/library/

Testing Scenarios

Scenario 1: Intersection

Created by	Sudha Vijayakumar
Creation Date	Mar 28, 2022
Test Scenario	#1
Scenario Type	External
Map	BorregasAve
Route	Vehicle driving southbound on Borregas Ave crosses the intersection at Caribbean Ave.
Scenario	A pedestrian crosses the crosswalk after the vehicle enters the intersection. Vehicle is in the right of way.
Description	1. Vehicle is driving southbound on Borregas Ave. 2. Vehicle crosses the limit lines and enters the intersection at Borregas and Caribean Ave. With intent of continuing on Borregas. 3. A pedestrian crosses the crosswalk (ignoring the no cross sign)
Expected Result	Vehicle slows down to grant the right of way to the pedestrian.
Actual Result	

Created by	Sudha Vijayakumar
Creation Date	Mar 28, 2022
Test Scenario	#2
Scenario Type	External
Map	BorregasAve
Route	Vehicle driving southbound on Borregas Ave crosses the intersection at Caribbean Ave.
Scenario	A 3rd party vehicle runs the redlight and will collide with the AV at 90 degrees.
Description	1. AV is driving southbound on Borregas Ave. 2. AV is given a green light and can enter the intersection 3. A 3rd party vehicle heading westbound on Caribbean Ave, runs the redlight.
Expected Result	AV should not enter the intersection. If it does it should perform an emergency break / acceleration to avoid collision.
Actual Result	

Scenario 2: Two lane road

Created by	Huyen Nguyen
Creation Date	Mar 28, 2022
Test Scenario	#1
Scenario Type	External
Map	BorregasAve
Route	Vehicle driving southbound on Borregas Ave after the Caribean Ave intersection onto the 2 lane street.
Scenario	There will be cones blocking off the only southbound lane of Borregas.
Description	1. AV is driving southbound on Borregas Ave. 2. There are cones blocking off the remainder of Borregas Ave after the single dotted yellow line.
Expected Result	AV should slow down, identify that it is single yellow lines and perform a pass in the opposite direction when safe to do so.
Actual Result	

Created by	Huyen Nguyen
Creation Date	Mar 28, 2022
Test Scenario	#2
Scenario Type	External
Map	BorregasAve
Route	Vehicle driving southbound on Borregas Ave after the Caribean Ave intersection onto the 2 lane street.
Scenario	There will be a vehicle stopped on Borregas Ave after the intersection.
Description	1. AV is driving southbound on Borregas Ave. 2. There is a stalled vehicle after the dotted yellow line on Borregas.
Expected Result	AV should slow down, identify that it is single yellow lines and perform a pass in the opposite direction when safe to do so.
Actual Result	

Senario 3: Parking Lot Exit

Created by	Richard Ngo
Creation Date	Mar 28, 2022
Test Scenario	#1
Scenario Type	External
Map	BorregasAve
Route	Vehicle is in the parking lot of: 1380 Borregas Ave and intends to exit and head North bound. (Right turn to exit)
Scenario	Exiting out of a parking lot with traffic.
Description	1. AV is exiting the parking lot at the north most exit for the building: 1380 Borregas Ave. 2. A box truck will cross in front of the exit driveway as the AV is approaching the ramp.
Expected Result	AV should slow down and stop if needed. After the truck has pass it should proceed with the right hand turn onto Borregas Ave (North Bound).
Actual Result	

Created by	Richard Ngo
Creation Date	Mar 28, 2022
Test Scenario	#2
Scenario Type	External
Map	BorregasAve
Route	Vehicle is in the parking lot of: 1394 Borregas Ave and intends to exit and head east bound on Caribbean Ave. (outermost lane)
Scenario	Exiting out of a parking lot with traffic.
Description	1. AV is exiting the parking lot at the north most exit for the building: 1394 Borregas Ave. 2. There are a series of vehicles there waiting for a red light to change. 3. The vehicles are blocking the exit to the parking lot
Expected Result	AV should stop at the exit ramp and wait until there is enough space for the AV to enter the road and turn right.
Actual Result	

Scenario 4: San Francisco city driving

Testing Scenarios	
Created by	Prit Dineshbhai Lakhani
Creation Date	Mar 28, 2022
Test Scenario	#5
Scenario Type	External
Map	SF
Route	Bob drives on the straight line
Scenario	The bob recognize the road block (road work in progress) and make decision according to the scene
Description	1. The Bob moves on the straight line 2. Bob recognizes a road block (road work in progress) on the road. 3. The Bob responds to the road blocking scene.
Expected Result	The Bob slows down the car and if there is no car behind the bob then after taking the side, Bob again drives normally.
Actual Result	

Testing Scenarios

Created by	Prit Dineshbhai Lakhani
Creation Date	Mar 28, 2022
Test Scenario	#6
Scenario Type	External
Map	SF
Route	Bob drives on the straight line.
Scenario	The bob recognizes the pedestrian and makes a decision according to the scene.
Description	1. The Bob moves on the straight line 2. Bob recognizes a pedestrian crossing the road. 3. In response to the scene Bob stops and waits.
Expected Result	The Bob detected the pedestrian crossing the road, in response Bob stops and waits for the pedestrian to cross the road.
Actual Result	

AB Testing Scenarios and Scripts

1. https://github.com/sudha-vijayakumar/CMPE287_AVT_SVL_SIMULATION/blob/main/Deliverable-1/simulator-json.zip (SCRIPTS)
2. Test Scenarios Recording - SVL Simulator (Video)

REFERENCES

[1] Koopman, P. and Wagner, M., "Challenges in Autonomous Vehicle Testing and Validation," *SAE Int. J. Trans. Safety* 4(1):15-24, 2016, https://doi.org/10.4271/2016-01-0128.

http://av-test-challenge.org/pdfs/2021%20IEEE%20AD%20AI%20Test%20Challenge%20SVL%20Simulator%202021.1.pdf

https://www.svlsimulator.com/docs/getting-started/getting-started/#autonomous-software-tutorials

https://arxiv.org/pdf/2005.03778v1.pdf