Edge Cases and Autonomous Vehicle Safety

SATURN, Pittsburgh PA
May 7, 2019
Making safe robots
- Doer/Checker safety

Edge cases matter
- Robust perception matters

The heavy tail distribution
- Fixing stuff you see in testing isn’t enough

Perception stress testing
- Finding the weaknesses in perception

UL 4600: autonomy safety standard
98% Solved For 20+ Years

- Washington DC to San Diego
  - CMU Navlab 5
  - Dean Pomerleau
  - Todd Jochem

- AHS San Diego demo Aug 1997

TRIP COMPLETE !!!
2797/2849 miles (98.2%)
NREC: 30+ Years Of Cool Robots

1985
ARPA Demo II
NASA Dante II
Auto Harvesting

1990
DARPA Grand Challenge
DARPA SC-ALV
NASA Lunar Rover
Auto Excavator

1995
Auto Forklift
DARPA PerceptOR

2000
DARPA LAGR
Mars Rovers
Urban Challenge

2005
Army FCS
Auto Haulage

2010
DARPA UPI
Auto Spraying
Laser Paint Removal

Carnegie Mellon University Faculty, staff, students
Off-campus Robotics Institute facility

Software Safety
Before Autonomy Software Safety

- The Big Red Button era
Traditional Validation Vs. Machine Learning

- Use traditional software safety where you can

..BUT..

- Machine Learning (inductive training)
  - No requirements
    - Training data is difficult to validate
  - No design insight
    - Generally inscrutable; prone to gaming and brittleness
APD (Autonomous Platform Demonstrator)

TARGET GVW: 8,500 kg
TARGET SPEED: 80 km/hr

Safety critical speed limit enforcement
Safety Envelope Approach to ML Deployment

- Specify unsafe regions
- Specify safe regions
  - Under-approximate to simplify
- Trigger system safety response upon transition to unsafe region
Architecting A Safety Envelope System

“Doer” subsystem
- Implements normal, untrusted functionality

“Checker” subsystem – Traditional SW
- Implements failsafes (safety functions)

Checker entirely responsible for safety
- Doer can be at low Safety Integrity Level
- Checker must be at higher SIL

(Also known as a “safety bag” approach; also monitor/actuator pair)
Validating an Autonomous Vehicle Pipeline

Perception presents a uniquely difficult assurance challenge.

Machine Learning Based Approaches

Randomized & Heuristic Algorithms

Run-Time Safety Envelopes

Doer/Checker Architecture

Control Systems

Control Software Validation

Doer/Checker Architecture

Autonomy Interface To Vehicle

Traditional Software Validation
Brute Force AV Validation: Public Road Testing

- Good for identifying “easy” cases
  - Expensive and potentially **dangerous**

Validation Via Brute Force Road Testing?

- If 100M miles/critical mishap...
  - Test 3x–10x longer than mishap rate
  - Need 1 Billion miles of testing

- That’s ~25 round trips on every road in the world
  - With fewer than 10 critical mishaps
Closed Course Testing

- Safer, but expensive
  - Not scalable
  - Only tests things you have thought of!
Simulation

- Highly scalable; less expensive
  - Scalable; need to manage fidelity vs. cost
  - Only tests things you have thought of!

Udacity


ANSYS
What About Edge Cases?

- Gaps in training data can lead to perception failure
  - Safety needs to know: “Is that a person?”
  - Machine learning provides: “Is that thing like the people in my training data?”

- Edge Case are surprises
  - You won’t see these in training or testing
    ➔ Edge cases are the stuff you didn’t think of!

<table>
<thead>
<tr>
<th>PREDICTED CONCEPT</th>
<th>PROBABILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>bird</td>
<td>0.997</td>
</tr>
<tr>
<td>no person</td>
<td>0.990</td>
</tr>
<tr>
<td>one</td>
<td>0.975</td>
</tr>
<tr>
<td>feather</td>
<td>0.970</td>
</tr>
<tr>
<td>nature</td>
<td>0.963</td>
</tr>
<tr>
<td>poultry</td>
<td>0.954</td>
</tr>
<tr>
<td>outdoors</td>
<td>0.936</td>
</tr>
<tr>
<td>color</td>
<td>0.910</td>
</tr>
<tr>
<td>animal</td>
<td>0.908</td>
</tr>
</tbody>
</table>

https://www.clarifai.com/demo

https://www.bit.ly/2In4rzj
Novel objects (missing from zoo) are triggering events


https://goo.gl/J3SSyu
Why Edge Cases Matter

Where will you be after 1 Billion miles of validation testing?

Assume 1 Million miles between unsafe “surprises”

- Example #1:
  100 “surprises” @ 100M miles / surprise
  - All surprises seen about 10 times during testing
  - With luck, all bugs are fixed

- Example #2:
  100,000 “surprises” @ 100B miles / surprise
  - Only 1% of surprises seen during 1B mile testing
  - **Bug fixes give no real improvement** (1.01M miles / surprise)

[Link to Edge Case Research](https://goo.gl/3dzguf)
Real World: Heavy Tail Distribution(?)

Common Things Seen In Testing

Edge Cases Not Seen In Testing

Random Independent Arrival Rate (exponential)

Power Law Arrival Rate (80/20 rule) (Heavy Tail Distribution)

Many Different, Infrequent Scenarios Total Area is the same!
Need to find “Triggering Events” to inject into sims/testing

The Heavy Tail Testing Ceiling

- Fault Injected Training
- Deployment Safety Goal
- Heavy Tail Ceiling
- "Hard Scenario" Weighted Training
- Brute Force Training
- Here There Be Dragons! (Unknowable Unknowns)
- Unique Surprises (Mostly Unseen in Training)
- Rare Events (Seen Once in Training)
- Unusual Events
- Everyday Events
Need to collect surprises
- Novel objects
- Novel operational conditions

Corner Cases vs. Edge Cases
- Corner cases: infrequent combinations
  - Not all corner cases are edge cases
- Edge cases: combinations that behave unexpectedly

Issue: novel for person ≠ novel for Machine Learning
- ML can have “edges” in unexpected places
- ML might train on features that seem irrelevant to people
Sensor data corruption experiments

Defocus & haze are a significant issue

Exploring the response of a DNN to environmental perturbations from “Robustness Testing for Perception Systems,” RIOT Project, NREC, DIST-A.

Gaussian Blur & Gaussian Noise cause similar failures
Brittle perception behavior indicates Edge Cases

- Can uncover false negatives and detect novel objects

**Hologram Detects Edge Cases**

**Diagram**

- Sensor Data
- Hologram Data Augmentation
- Customer Perception Algorithm
- Customer Perception Algorithm
- Hologram Analysis
- "Surprise" Sensor Sequences

**Edge Cases:**
- False negatives
- Novel objects
- False positives
- Combinations
Context-Dependent Perception Failures

- Perception failures are often context-dependent
  - False positives and false negatives are both a problem

Will this pass a “vision test” for bicyclists?
Example Triggering Events via Hologram

- Mask-R CNN: examples of systemic problems we found

Notes: These are baseline, un-augmented images // Your mileage may vary.
Drivers do more than just drive
- Occupant behavior, passenger safety
- Detecting and managing equipment faults

Operational limitations & situations
- System exits Operational Design Domain
- Vehicle fire or catastrophic failure
- Post-crash response

Interacting with non-drivers
- Pedestrians, passengers
- Police, emergency responders
Handling updates
- Fully recertify after every weekly update?
- Security in general

Vehicle maintenance
- Pre-flight checks, cleaning
- Corrective maintenance

Supply chain issues
- Quality fade
- Supply chain faults

Is windshield cleaning fluid life critical?
Safety Standard Landscape

IEEE P700x
- Ethical Life-Cycle

ISO 21448 (SOTIF):
- Enumerate Scenarios
- Triggering Events
- Historically for ADAS

ISO 26262:
- Functional Safety
- Equipment Faults
- Design Faults
- Controllability
Note: ISO 26262 covers functional safety lifecycle activities.

UL 4600: (Draft Scope)
- Machine Learning Faults
- External Operational Faults
- Non-Driver Human Faults
- Nondeterminism
- High Residual Unknowns
- Field Feedback and Updates
- Lack of Human Oversight
- System-Level Safety Metrics
- Full Autonomous Driving: Excludes Human Supervision

Note: Provides significantly more detailed treatment regarding autonomous systems for those topics also mentioned in ISO 21448, ISO 26262. Intended to supplement, not replace these.
Ways To Improve AV Safety

- More safety transparency
  - Independent safety assessments
  - Industry collaboration on safety

- Minimum performance standards
  - “Driver test” is necessary -- but not sufficient
    - How do you measure maturity?

- Autonomy software safety standards
  - ISO 26262/21448 + UL 4600 + IEEE P700x
  - Dealing with uncertainty and brittleness