Using Machine Learning Safely in Automotive Software: An Assessment and Adaption of Software Process Requirements in ISO 26262

Rick Salay, Krzysztof Czarnecki
[https://arxiv.org/abs/1808.01614]

Presenter: Claudia Athens

Introduction

- Usage of ML in automotive software development is rising
- Safety is the critical objective in automotive industry
- Standards (ISO 26262) help industry approach safety in a consistent manner

What is ISO 26262?

- Regulates functional safety of road vehicles
- Part 6
  - “Product development at the software level”
  - Specifies compliance requirements for software development process
  - Details methods needed in the development process

What does ISO 26262 Part 6 cover?

Category of Method	Number of Methods	Description
Coding guidelines	8	Coding standards to improve consistency and comprehension.
Architecture notations	3	Degrees of formality in design notation.
Architecture design	7	Design best practices to manage complexity.
Architecture error detection	6	Error detection methods for fault tolerance.
Architecture error handling	4	Error recovery methods for fault tolerance.
Architecture verification	7	Methods of verification against safety requirements.
Unit design notations	4	Degrees of formality in design notation.
Unit design and implementation	10	Design and coding best practices to manage complexity.
Unit design and implementation verification	8	Methods of verification against safety requirements.
Unit testing	5	Types of unit testing.
Unit deriving test cases	4	Deriving test cases from requirements.
Unit testing coverage metrics	3	Code coverage of test cases.
Integration testing	5	Types of integration testing.
Integration deriving test cases	4	Deriving test cases from requirements.
Integration testing coverage metrics	2	Architecture coverage of test cases.
Verification of software safety requirements	3	System level testing to ensure the embedded software satisfies safety requirements.

Problem and goal

- ISO 26262 was not designed with ML in mind
  - Balancing desire to innovate with primary objective of safety

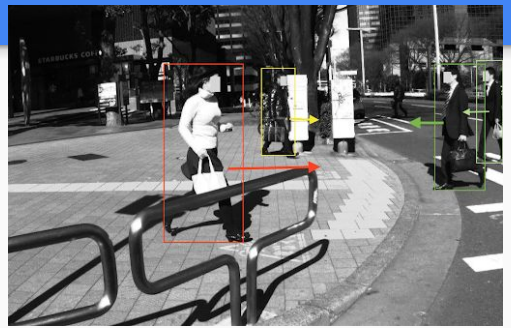
- Address conflicts between standard and ML such that it can be applied more widely

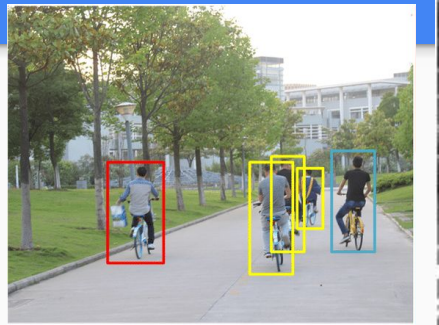
Safety in software

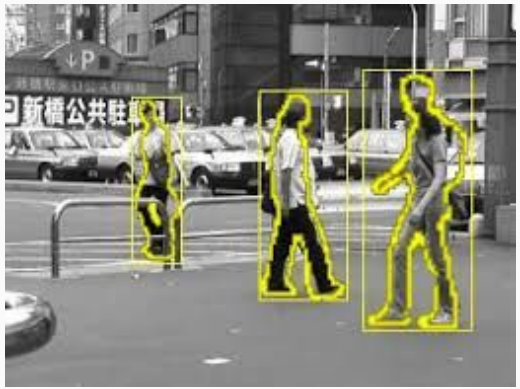
- Safety is the “absence of unreasonable risk”
- Safety assurance principle
  - “By developing software using an adequate level of rigor, the residual risk of hazard due to software failure can be reduced to an acceptable level”

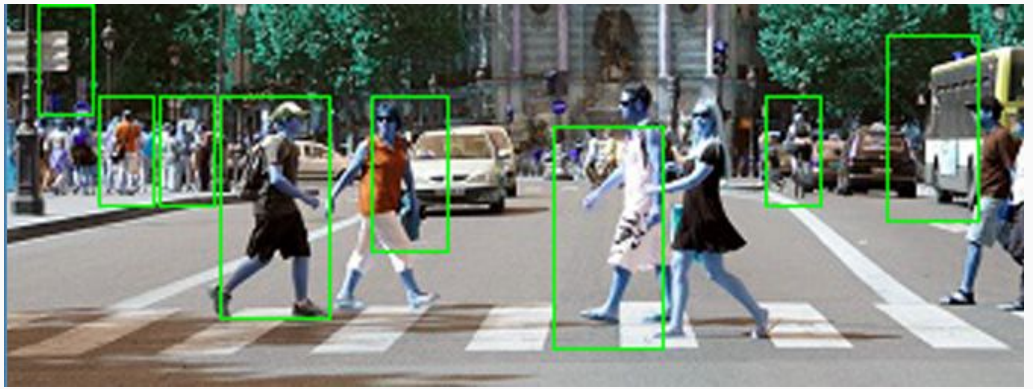
- Obstacles to SAP with ML
  - Lack of specification
  - Non-interpretability

What is the specification for recognizing a pedestrian?







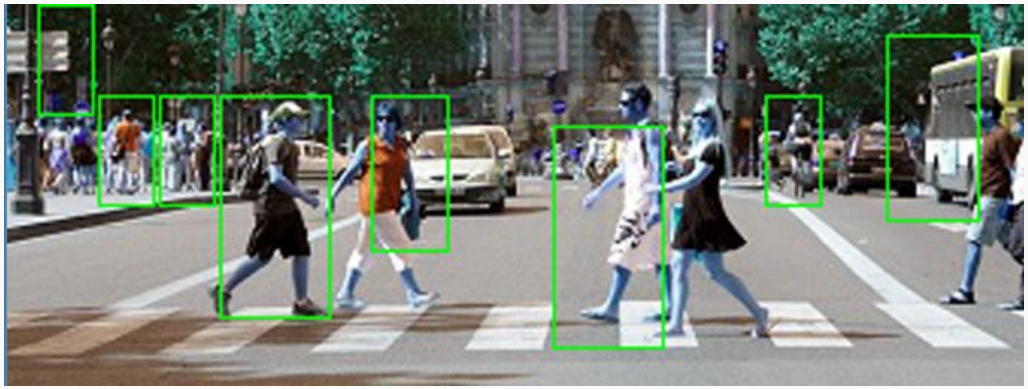


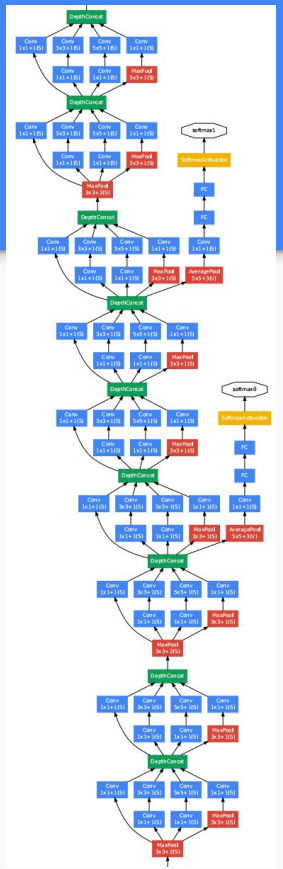
Obstacle 1: Lack of specification

- Environments and tasks in ML are not fully specifiable
- Using training to teach examples rather than fully specified rules
- But training sets are risky
  - No guarantees it will represent inputs well
  - High risk situations may be not included

Obstacle 2: Non-interpretability

- ML models are not easily interpretable
- Results can be difficult to explain





ISO 26262 analysis approach

- Authors seek to fit the requirements of ISO 26262 to ML-based software
  - Requirements are reinterpreted for ML-based software
  - Additional requirements are suggested to fill gaps
  - Try to mitigate the obstacles

Recommendation 1:
Add an ML decision gate (O1)

- “An assessment shall be performed to determine whether the safety requirement must be implemented by an ML-component or can acceptably be implemented using a programmed component. If the latter case holds then programming shall be used rather than ML”

- Ex: “detect all pedestrians within 10 meters” vs “detect obstacles within 10 meters”
  - Splitting the component into a programmed and an ML part

Recommendation 2:
Split implementation best practices (O2)

- Not all design principles make sense in the context of ML
- ML needs its own best practices

	Method
a	One entry and one exit point in subprograms and functions
b	No dynamic objects or variables, or else online test during their creation
c	Initialization of variables
d	No multiple use of variable names
e	Avoid global variables or else justify their usage
f	Limited use of pointers
g	No implicit type conversions
h	No hidden data flow or control flow
i	No unconditional jumps
j	No recursions

Recommendation 3:
Selective feature selection (O2)

- New requirement:
  - “An analysis shall be performed to show that all features used by the ML model are causally related to the output of the ML component.“
  - Ex: Detecting a pedestrian by looking at a pixel value versus looking at the size of the object

Recommendation 4:
Augment error handling for ML (O1)

- Based on the required level of safety in a system, different error handling is required by ISO 26262

- Additional proposed methods for ML
  - Data harvesting
  - Redundant classifiers

	Method
a	Static recovery mechanism
b	Graceful degradation
c	Independent parallel redundancy
d	Correcting codes for data

Why do these standards matter?

- Standards make collective industries better
- Organizations come together for guidelines across the industry
- Customers and companies benefit

- Nonfunctional requirements of a project to comply with standards
- Standards can also drive functional requirements of systems
- Getting the right requirements

Where to go next?

- This work is just the tip of the iceberg for adapting requirements for ML
  - More research is needed
- These recommendations can be applied to other standards/industries
- Once we have standards on how to write the software, we can make better standards on what the software should do

Where to go next?

- Get industry support to amend the ISO 26262 standard
  - Start with author's recommendations
  - Keep up with recent advancements in the field
  - Hold organizations to this standard
  - Progress ML development with standards

Conclusion

- ML does not traditionally mix well with requirements related to safety
- Additions and changes to make the ISO 26262 better comply with ML
- Applying the recommended requirements can reduce the risk and increase the safety of using ML in software

Further Reading

Salay, Rick, and Krzysztof Czarnecki. "Using machine learning safely in automotive software: An assessment and adaption of software process requirements in ISO 26262." *arXiv preprint arXiv:1808.01614* (2018).

Salay, Rick, Rodrigo Queiroz, and Krzysztof Czarnecki. "An analysis of ISO 26262: Using machine learning safely in automotive software." *arXiv preprint arXiv:1709.02435* (2017).

K. R. Varshney, "Engineering safety in machine learning," *2016 Information Theory and Applications Workshop (ITA)*, La Jolla, CA, 2016, pp. 1-5, doi: 10.1109/ITA.2016.7888195.

Questions