

“Engineerable AI” Project

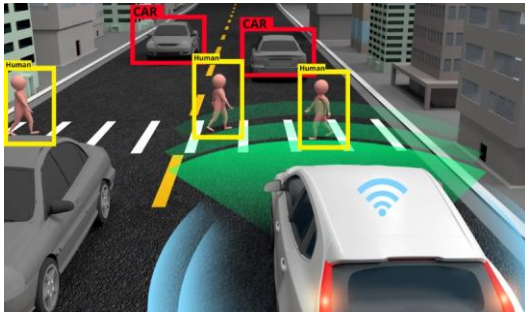
Engineerable AI Techniques for
Practical Applications of High-Quality
Machine Learning-based Systems

Fuyuki ISHIKAWA

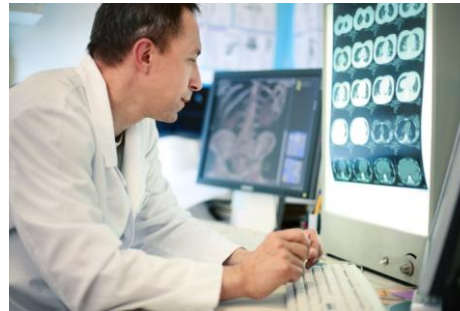
National Institute of Informatics, Japan

Background

- Active investigation for industrial applications of machine learning (ML)-based AI systems
- ➔ **Increasing demands for quality assurance**
 - Not only “higher accuracy for the given dataset”
 - Quality-sensitive application domains and customers



Autonomous Driving



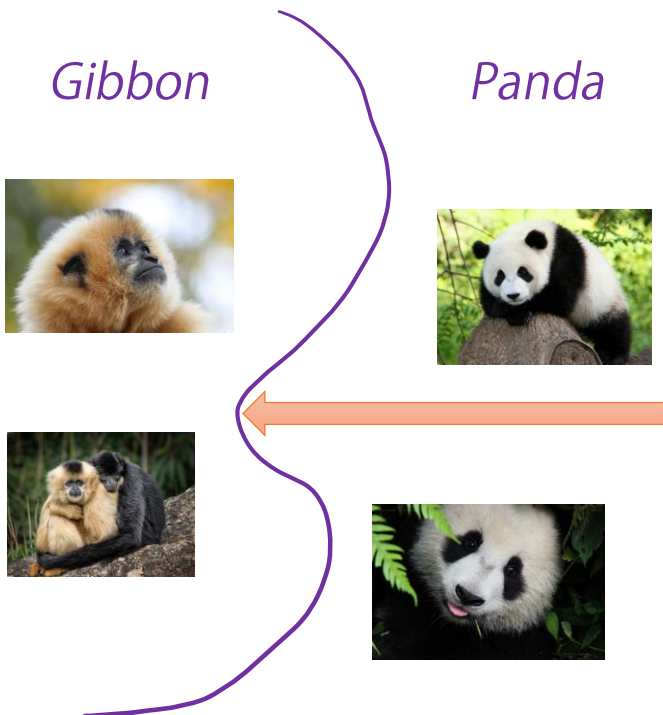
Medical Diagnosis



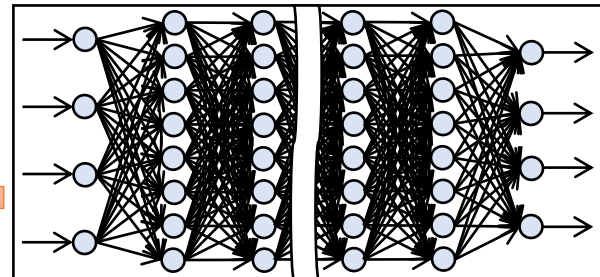
Process Automation

Background

- Unique difficulties in ML :
data-driven behavior and uncertainty



Software 2.0 as a collection of enormous parameters in deep neural networks



- *Unclear boundary of functionality*
- *Unpredictability of quality to be achieved*
- *Weakness against noisy inputs*
- ...

Behavior (classification this time) is generated from training data

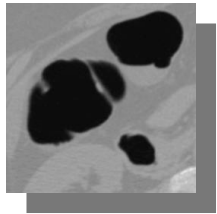
[<http://free-photos.gatag.net/>]

Problems

From discussions/surveys with 200+ practitioners

Medical Diagnosis

Want AI to detect cases overlooked by human doctors!



Failure with polyps at specific locations

Difficult with very few data!

Rare cases more difficult for AI

Years for collecting data necessary in using deep learning



Autonomous Driving

Want to ensure safety of AI under various situations!

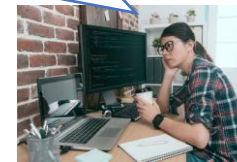


Failure with pedestrians under specific situations

Difficulty with unstable assurance activities!

Enormous target situations

Trial-and-error for fixing something, then others broken



Our Vision

■ “Hard to Engineer” problems, why?

By building the whole functionality only with data

- Dependency on large data
- Uncontrollability of detailed behavior

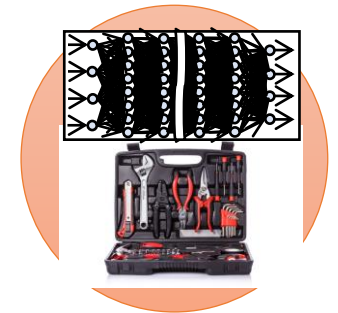
➔ Towards “Engineerable AI”

Effectively tailored for quality requirements

By incorporating “knowledge” not only data

- Knowledge-incorporated construction
- Knowledge-driven improvement

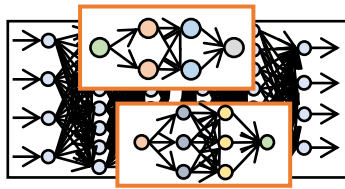
AI Techniques
+
Software Engineering



Technical Approach

(1) Knowledge-incorporated construction

- Design models to reflect domain requirements



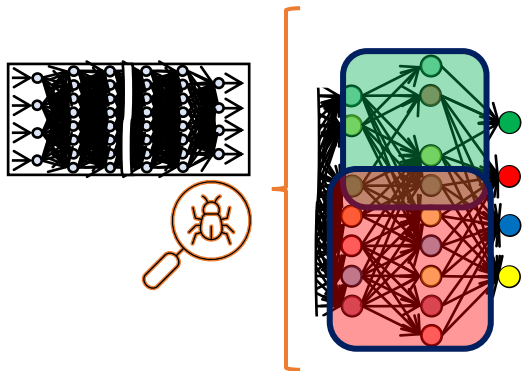
Example:

design modules to deal with individual types of cases

*Achieve high performance with fewer data
with more controllability and accountability*

(2) Knowledge-driven debugging

- Improve models by identifying causes of success/failure



Example:

identify parts responsible for successful behavior and those for undesirable behavior

*Controlled fix for targeted issues
without unexpected degradation*

Proof-of-Concept

(1) Knowledge-incorporated construction

- Design models to reflect domain requirements

Medical
Diagnosis



Want AI to detect cases overlooked by human doctors!



Done: higher performance diagnosis with only 100 data

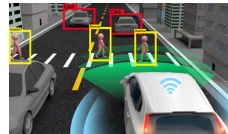


Difficult cases with less data

(2) Knowledge-driven debugging

- Improve models by identifying causes of success/failure

Autonomous
Driving



Want to ensure safety of AI under various situations!

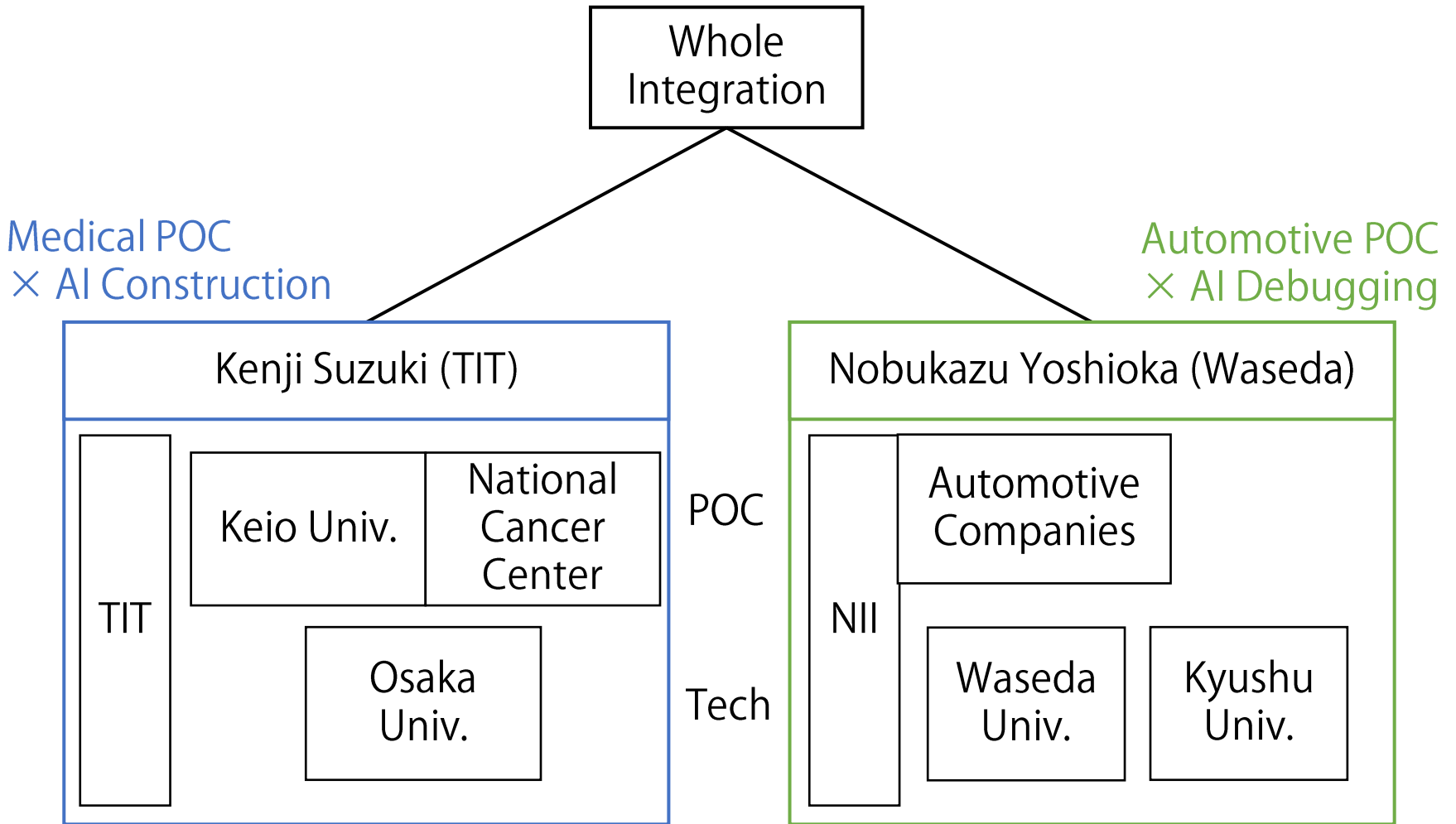


Done: controlled fix with little degradation in significant situations



Safety with enormous situations

Project Teams



Summary

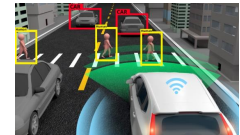
- Engineerable AI project
 - For high-quality ML-based systems
 - Tackling difficulties in dependency on large data and uncontrollability of deep learning

Medical
Diagnosis



Want AI to detect cases overlooked by human doctors!

Autonomous
Driving



Want to ensure safety of AI under various situations!

Engineerable AI techniques

Knowledge-incorporated construction
Knowledge-driven debugging