2022 年 7 月 12 日 18:00 作成

機械学習品質マネジメント
リファレンスガイド

2022 年 7 月 14 日

国立研究開発法人産業技術総合研究所

デジタルアーキテクチャー研究センター
テクニカルレポート DigiARC-TR-2022-04

サイバーフィジカルセキュリティ研究センター
テクニカルレポート CPSEC-TR-2022005

人工知能研究センター
テクニカルレポート

© 2022 国立研究開発法人産業技術総合研究所

前書き

本リファレンスガイドは、国立研究開発法人産業技術総合研究所の機械学習品質マネジメント検討委員会において作成した。

本文書の作成は、国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の委託事業（JPNP20006）の一環として行われたものである。

目次

1 はじめに 1
2 目的 2
3 アプローチ 3
4 本リファレンスガイドの構成 4
5 ガイドラインの概要 5
6 品質アセスメントシート 6
6.1 概要 6
6.2 AI 開発プロセス 6
6.3 品質アセスメントシート 9
6.3.1 システム要求分析票 10
6.3.2 システム・リスクアセスメント票 11
6.3.3 AI 要求分析票 13
6.3.4 データセット・アセスメント票 14
6.3.5 学習モデル・アセスメント票 15
6.3.6 保守計画アセスメント票 16
6.3.7 機能安全開発 17
6.4 開発プロセスおよびアセスメントシートの適用例 18
6.4.1 適用される AI の内容 19
6.4.2 システム要求分析 20
6.4.3 システムリスクアセスメント 20
6.4.4 AI 要求分析 21
6.4.5 データセットの設計と収集 21
6.5 まとめ 23
7 自動運転車 24
7.1 概要 24

7.2 ビジネス要件 24
7.2.1 製品名 24
7.2.2 ユースケース 25
7.2.3 背景 25
7.2.4 目的・目標 25
7.2.5 製品のステークホルダー 25
7.2.6 ステークホルダーの初期要求 26
7.2.7 ビジネス要件の詳細 26
7.2.8 外部品質に関する要求事項 29
7.2.9 外部品質特性レベルを定義する 30
7.2.10 おわりに 30
7.3 問題の予備的分析 30
7.3.1 技術仕様 31
7.3.2 安全仕様 31
7.3.3 KPI 仕様 32
7.4 PoC（Proof of Concept）フェーズ 33
7.4.1 既存データセットの予備調査 33
7.4.2 選択したデータセットの紹介 34
7.4.3 データの分布 35
7.4.4 候補モデルの事前訓練 37
7.4.5 PoC フェーズで得られた知見 39
7.5 内部品質評価を伴う AI 開発 40
7.5.1 A-1: 問題領域分析の十分性 40
7.5.2 A-2: データ設計の十分性 46
7.5.3 B-1: データセットの被覆性 51
7.5.4 B-2: データセットの均一性 56
7.5.5 B-3: データの妥当性 60

7.5.6 C-1: 機械学習モデルの正確性 61
7.5.7 C-2: 機械学習モデルの安定性 65
7.5.8 D-1: プログラムの信頼性 80
7.5.9 E-1: 運用時品質の維持性 82
7.6 用語集 89
8 金属鋳物の外観検査 91
8.1 ビジネス要件 91
8.1.1 背景 91
8.1.2 目的・目標 91
8.1.3 AI システムのステークホルダー 91
8.1.4 ステークホルダーの初期要求 91
8.1.5 ビジネス要件の詳細 92
8.1.6 外部品質に関する要求事項 92
8.1.7 達成すべき外部品質特性レベルの特定 93
8.2 品質マネジメントの手順 93
8.2.1 A-1: 問題領域分析の十分性 94
8.2.2 A-2: データ設計の十分性 96
8.2.3 B-1: データセットの被覆性 98
8.2.4 B-2: データセットの均一性 100
8.2.5 B-3: データの妥当性 101
8.2.6 C-1: 機械学習モデルの正確性 101
8.2.7 C-2: 機械学習モデルの安定性 104
8.2.8 D-1: プログラムの信頼性 107
8.2.9 E-1: 運用時品質の維持性 107
9 郵便番号の分析 109
9.1 はじめに 109
9.2 ビジネス要件 109

9.2.1 問題定義（ユースケース） 109
9.2.2 背景 110
9.2.3 目的・目標 110
9.2.4 この製品のステークホルダー 110
9.2.5 ステークホルダーの初期要求 110
9.2.6 ビジネス要件の詳細 111
9.2.7 外部品質に関する要求事項 111
9.2.8 外部品質特性レベルを定義する 112
9.2.9 おわりに 112
9.3 製品仕様 113
9.3.1 データ関連仕様 113
9.3.2 モデル仕様 113
9.3.3 KPI 仕様 113
9.4 データセットの紹介 114
9.4.1 データセットの探索 114
9.4.2 MNIST データセット 114
9.4.3 入力データのサンプル 114
9.5 MLQM ガイドラインを用いた品質保証手順 115
9.5.1 A-1：問題領域分析の十分性 115
9.5.2 A-2: データ設計の十分性 120
9.5.3 B-1: データセットの被覆性 121
9.5.4 B-2: データセットの均一性 130
9.5.5 B-3: データの妥当性 134
9.5.6 C-1: 機械学習モデルの正確性 135
9.5.7 C-2: 機械学習モデルの安定性 137
9.5.8 D-1：プログラムの信頼性 141
9.5.9 E-1：運用時品質の維持性 144

10 住宅価格分析 148
10.1 はじめに 148
10.2 ビジネス要件の詳細 148
10.2.1 ユースケース 148
10.2.2 背景 148
10.2.3 目的・目標 148
10.2.4 この製品のステークホルダー 149
10.2.5 ステークホルダーの初期要求 149
10.2.6 ビジネス要件の詳細 149
10.2.7 外部品質に関する要求事項 150
10.2.8 外部品質特性レベルを定義する 150
10.2.9 おわりに 151
10.3 製品仕様 151
10.3.1 モデル仕様 151
10.3.2 データ関連仕様 151
10.3.3 KPI 仕様 151
10.4 データセットの紹介 151
10.4.1 データセットの探索 151
10.4.2 入力データのサンプル 152
10.5 MLQM ガイドラインを用いた品質保証手順 152
10.5.1 A-1：問題領域分析の十分性 152
10.5.2 A-2: データ設計の十分性 156
10.5.3 B-1: データセットの被覆性 157
10.5.4 B-2: データセットの均一性 159
10.5.5 B-3: データの妥当性 160
10.5.6 C-1: 機械学習モデルの正確性 160
10.5.7 C-2: 機械学習モデルの安定性 164

10.5.8 D-1: プログラムの信頼性 166
10.5.9 E-1: 運用時品質の維持性 169
11 自動搬送車 171
11.1 製品名 171
11.2 ユースケース 171
11.3 ビジネス要件 172
11.3.1 背景 172
11.3.2 目的・目標 172
11.3.3 この製品のステークホルダー 173
11.3.4 ステークホルダーの初期要求 173
11.3.5 ビジネス要件の詳細 173
11.3.6 外部品質に関する要求事項 175
11.3.7 外部品質のレベルを定義する 176
11.4 おわりに 176
付録 177
A. ビジネス要件記述 177
A.1 ステークホルダーの選択 177
A.2 繰返しによる調整 177
A.3 視点の選択 177
A.4 追加システムの要件 178
A.5 フォーマットの選択 178
B. Surprise Adequacy 179
C. 1 ピクセル変更 182
D. 評価の要約 184
D.1 自動運転車 184
D.2 金属鋳物の外観検査 188
D.3 郵便番号の分析 191

D.4　住宅価格分析 194
E.　品質アセスメントシート 197
参考文献 216
編集者・執筆者 221
改版履歴 223

1 はじめに

AI システムは、ここ 10 年ほどの間にその目覚しい性能の高さで注目を集め、今後、世界中の社会のさまざまな分野で幅広く導入されることが期待されている。そのような AI システムの品質は重要である。今後、AI システムは重要な場面でも利用され、その動作が適切でないと重大な結果を招くことが予期されるからである。

しかし、AI システムの品質には、従来のソフトウェア品質管理手法が通用しない側面がある。そこで、機械学習品質マネジメント検討委員会が作成した「機械学習品質マネジメントガイドライン」 [1]（以下、「MLQM ガイドライン」または単に「ガイドライン」という）では、AI システムの品質管理に適したアプローチを提示している。このガイドラインでは、品質目標を設定する際の視点と、目標達成のために AI ライフサイクルにおいて実践するべき重要な事項を示している。

ガイドラインは、AI 品質管理手法の基本的な考え方を示しているが、具体的な AI システムへの適用方法の詳細は記載していない。しかし、高品質な AI システムを開発・維持しようとする技術者にとってそのような詳細は不可欠である。本リファレンスガイドは、その必要に応えることを狙いとしている。

2 目的

このリファレンスガイドは、ガイドラインを用いた品質管理の事例を示すことにより、AI システムエンジニアにガイドラインの使い方を理解してもらうことを目的としている。

本リファレンスガイドの執筆者は、開発受託者（サービス開発者）の立場に立っている。本リファレンスガイドは、サービス開発の技術的な出発点となり、初期段階からこれに沿って開発を進めることで、開発プロセス全体を通じて品質基準が維持されることを保証しようとするものである。

MLQM ガイドラインを AI 製品の開発プロセスに適用することにより、以下の効果が期待できる。

- ビジネス要件における品質目標を明確に表現できる
- AI 製品の設計・開発・運用時の品質を評価するための明確な基準が得られる
- AI モデルの誤動作による事故リスクを低減するための、安全上重要なシナリオが事前に特定できる
- 最終製品の品質をより適切に評価でき、目標品質の未達を検出できる
- 最終ユーザーに対し、製品の品質、安全性、信頼性をわかりやすく提示できる

3 アプローチ

本文書には、MLQM ガイドラインの利用法を説明するために、様々な分野の AI システムの事例として以下の 5 件を掲載した。

1. 自動運転向け物体検知および場面識別
2. 金属鋳物の外観検査
3. 数字識別による郵便番号の認識
4. 住宅価格予測
5. 無人搬送車

今回、これらの事例で開発した AI システムは架空の、実験的なものである。典型的な AI 製品の評価と品質保証に、MLQM ガイドラインがどのように適用できるかを示すためだけに開発した。事例によっては、MLQM ガイドラインが示す 3 つの外部品質のうち、ある特定の品質を強調していることもあるが、現実の世界では、これらの品質すべてが要件に従って適切に評価されなければならない。

最初の 4 つの例は、それぞれ 2 つの部分から構成されている。

I. 満たすべきビジネス要件

II. ガイドラインに沿った品質管理手順

なお、この版のリファレンスガイドでは、例 5 はパート I のみを含む。

これらの事例は、一人ではなく多くの研究者や技術者のさまざまなアイデアや意見を集約したものである。また、ガイドラインの主要な利用者である 2 つのグループのために、2 つの異なった視点から作成されている。パート I に示した**ビジネス要件記述**（BRD）（付録 A を参照）は、ある特定のサービスや用途のために AI を用いた製品を開発したいサービス提供者の観点で書かれている。ここでは、開発チームに最終的な製品の要求仕様を提示するために、架空の BRD を作成した。

MLQM ガイドラインが示す外部品質は、サービス提供者による品質目標や利用時品質の実現に役立つものである。そのため、各 BRD の最後に、ビジネス要件をガイドラインの外部品質で表現し、開発者にとって明確で理解しやすいものになるようにしている。同時に、最終製品が期待される品質要求からどの程度乖離しても許容されるかも示している。

次に、執筆者はサービス開発者もしくは開発受託者の役割を担い、MLQM ガイドラインを技術的な拠り所として、AI サービスを開発する。そこで、パート II では、MLQM ガイドラインの内部品質評価手順に従って、与えられた BRD に従った品質基準が、開発の初期段階からプロセス全体を通じて達成されているかどうかを評価する。

最後に、品質評価と管理の全プロセスを記録する必要がある。このリファレンスガイドでは、プロセスの記録を容易にする品質アセスメントシートを紹介する。

4 本リファレンスガイドの構成

本リファレンスガイドの以降の構成は以下の通りである。

第 5 章は、ガイドラインが提唱するアプローチを簡単にまとめたものである。

第 6 章は、AI 品質アセスメントシートについて、その役割、構成、構造を説明する。

第 7 章から第 11 章では、ガイドラインに沿った AI 品質管理の事例を紹介する。各章で扱う AI の特徴は、以下の通りである。

- 第 7 章は自動運転
- 第 8 章は金属鋳物の外観検査
- 第 9 章は郵便番号認識
- 第 10 章は住宅価格予測
- 第 11 章は無人搬送車

ガイドラインの内容にすでに馴染みのある読者は、第 7 章から第 11 章の事例をすぐに読み始めてもよい。ガイドラインが提唱している考え方やアプローチを知りたい方のためには、第 5 章にその概要を示した。AI 品質マネジメントをすでに実践している人は、その計画、整理、記録などのヒントを第 6 章で得ることができるかもしれない。

この版のリファレンスガイドの限界について、あらかじめ読者にいくつかお断りしておく。第一に、ここで扱った事例は架空のものであり、ビジネス要件に関する記述はしばしばかなり大雑把なものとなっている。少なくとも、目標とそれがなぜ重要かについては説明するよう努めた。第二に、本文書で説明されている品質管理の取り組みは、ビジネス要件から導かれる品質目標を達成するには不十分である。これは執筆者に利用できる有用なデータの量が限られていたためである。第三に、リスクアセスメントの例では、安全性ばかりに目が行きがちで、公平性の欠如やプライバシーの喪失など、他の種類のリスクには必ずしも十分な注意が払われていない。その他、不十分と思われる箇所には脚注を付け、手本となる記述でないことを示した。しかしながら、このような例でも、大まかな参考にはしていただけるものと思う。

5 ガイドラインの概要

MLQM ガイドラインでは、AI システムには機械学習要素が 1 つ以上使用されており、AI システムの利用時品質は機械学習要素の外部品質に依存し、外部品質は機械学習要素のライフサイクルにおける内部品質の達成と維持により実現されると想定している。

このような前提のもと、ガイドラインでは、AI システムの品質管理について、以下のプロセスを提唱している。

1. AI システムの利用時品質に関する要件を確認する。
2. システム内の機械学習要素に対する外部品質要件を特定する。
3. 外部品質の要求レベルを決定する。
4. 各内部品質について、ガイドラインに記載されている要求レベルに対応する要件を検索する。
5. 測定と改善の繰り返しにより要件を満たすよう試みる。
6. 要件を満たそうとする過程とその結果を記録する。

ガイドライン第 2 版では、3 つの外部品質と 9 つの内部品質を挙げている。外部品質は以下の通りである。

- 安全性／リスク回避性
- AI パフォーマンス
- 公平性

内部品質は以下の通りである。

A-1 問題領域分析の十分性

A-2 データ設計の十分性

B-1 データセットの被覆性

B-2 データセットの均一性

B-3 データの妥当性

C-1 機械学習モデルの正確性

C-2 機械学習モデルの安定性

D-1 プログラムの信頼性

E-1 運用時品質の維持性

また、ガイドラインでは、AI システムの開発には反復プロセスが必要であり、システムの利用時品質に関する要求や AI 要素の外部品質の要求レベルは、数回の試行を通じて徐々に決定・調整されることが多いと想定している。その場合、各試行において、ガイドラインが提唱する上記のプロセスを実践することが望ましい。

6 品質アセスメントシート

本章では、AI 利用システムの開発におけるリファレンスガイドとして、具体的にどのように「機械学習品質マネジメントガイドライン」を活用できるかを検討し、その開発プロセス、および開発プロセスを支援する品質アセスメントシートを開発したので紹介する。このプロセスは、機能安全開発における概念をベースに拡張したものであるが、大筋は汎用になるよう、すなわち、AI 利用システムの様々な品質を管理するのに役立つよう設計した。

6.1 概要

本章の作成にあたり、まず AI の「リスク回避性」を確保するために、機能安全をベースとした開発の考え方を検討した。特に、システムの利用想定とそこで必要になる機能を明確化するとともに、リスク低減に向けた制御装置のハードウェア、ソフトウェアの設計を行い、AI モジュールへの要求仕様の明確化を行った。

次に、その結果から得られた要求仕様を達成するために、ガイドラインの内部品質に基づく AI 利用システムを実現する「開発プロセス」の提案を行った。

さらに、開発プロセスを実際に推進する上で、これを支援し、エビデンスとするための「品質アセスメントシート」を開発した。

6.2 AI 開発プロセス

図 1 に示すように、ガイドラインでは品質を、3 つに分けて理解する。第 1 は、システムがその全体として利用時に満たすことが期待される「利用時品質」。第 2 は、システムのうち機械学習で構築された構成要素が満たすことが期待される「外部品質」、第 3 は機械学習による構成要素が固有に持つ「内部品質」である。機械学習要素の「内部品質」の向上を通じてその「外部品質」を必要となるレベルで達成し、最終的なシステムの「利用時品質」を実現するものと整理している。そこには、安全性と同様に、有効性、公平性、などその他の性質も含まれる。

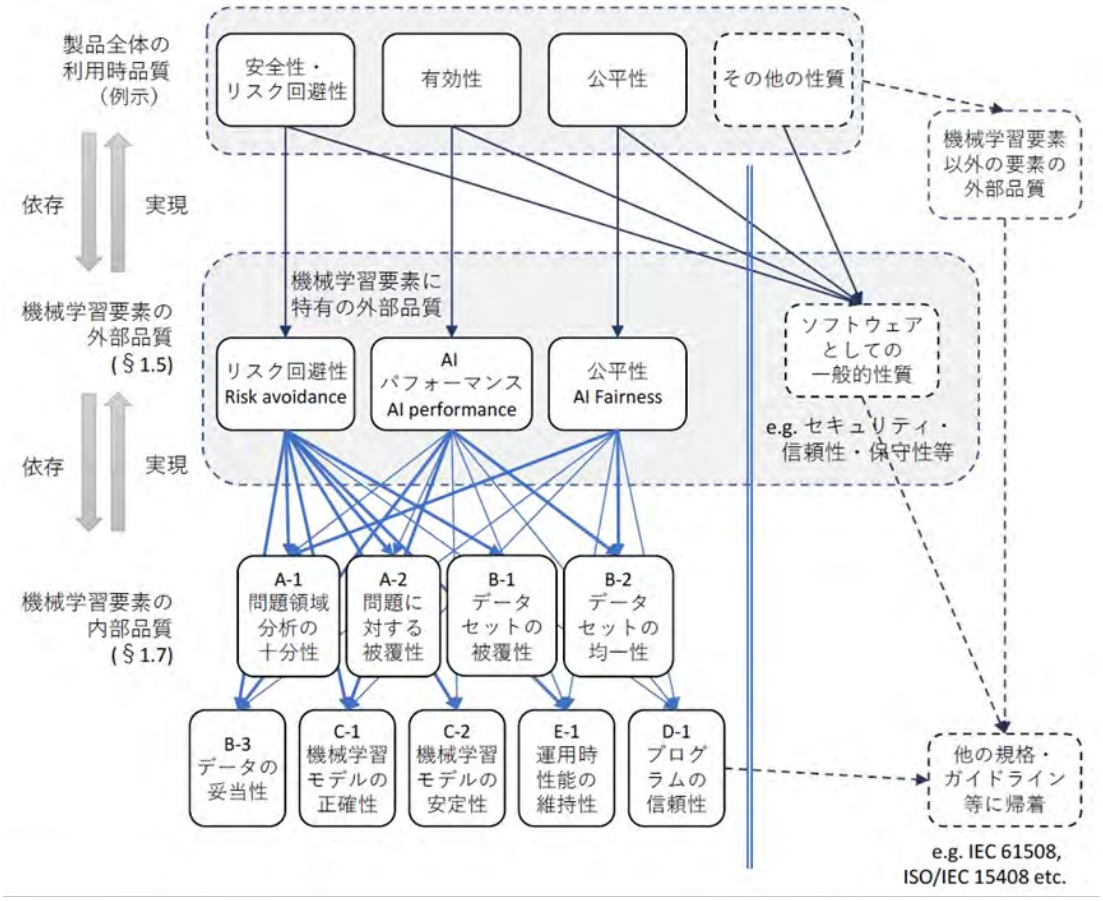


図 1 製品品質実現の全体構造

（「機械学習品質マネジメントガイドライン第２版」より）

本章において対象とする AI 利用システムは、ロボットなど制御で安全を担保する場合、そのシステムの開発や制御装置の設計は IEC61508 のような機能安全の規格に従う必要がある。本章執筆時点で明確に規定された安全規格は検討中ではあるものの、AI モジュールは、機能安全で取り扱われるソフトウェアもしくはプログラマブルな要素の一種であるため、機能安全の考え方をベースに開発プロセスの検討を行った。

また、開発プロセスの検討にあたっては、「機械学習品質マネジメントガイドライン」の品質実現の全体構造（図 1）にもとづいて、機械学習要素の「内部品質」に各開発プロセスを紐づける形で、機能安全の開発プロセスを拡張した（図 2)。そして、この開発プロセスを実践することで「内部品質」の実現を図り、最終的にターゲットとする製品全体の「利用時品質（安全性・リスク回避性)」の実現を目指す。

本章で提案する開発プロセスは、図 2 に示すようなプロセスで構成される。なお、各プロセスは開発における位置づけを示すものであり、開発順序は図 2 に示したフローに限定されるものではなく、繰り返しのループの入出力範囲も図に示したものは一例であり、開発の段階においてそれぞれ異なる。例えば、PoC(Proof of Concept)初期段階の場合、ターゲットシステムは未定で、AI モジュールのみの検討と実現性検証を行う場合などは、図 2 の右

側のみの開発プロセスで実施することもある。

図 2 は機能安全を主に機能安全のためのワークフローであり、そのように図中でも示しているが、ほぼ同様のフローが、AI ベースのシステムの他の品質を確保するためにも有用と期待している。今後、さらに検討し、追加の実験によって検証する必要がある。

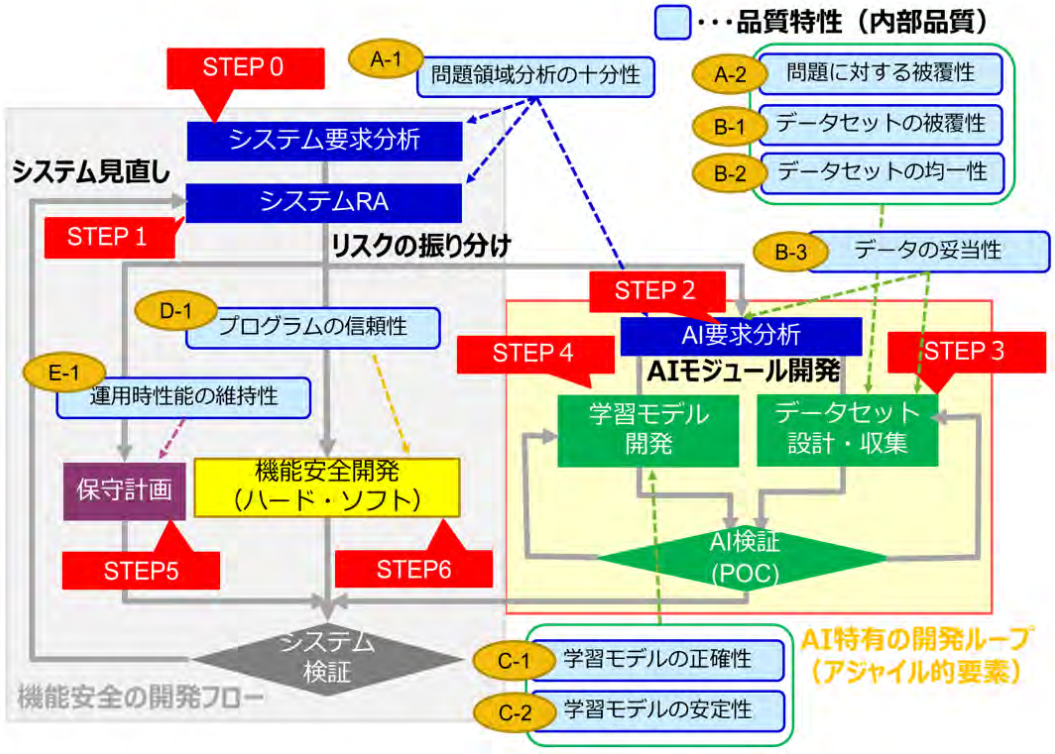


図 2 AI 特有の開発ループ

各プロセスでは次のような検討を行う。まず「システム要求分析」を行いシステムの要件定義を行い、次に「システム・リスクアセスメント（以下、RA）」によって、システムのリスクの洗い出しと、そのリスク低減方策の検討を行う。ここで、リスク低減方策を大きく 3 つに振り分ける。左から、システムの開発後も含めた保守・運用によりリスク低減を図るための「保守計画」、システムの AI モジュール以外のハードウェア・ソフトウェアの開発に関してリスク回避を達成する「機能安全開発」、そして AI 特有となる AI 設計を含む「AI モジュールの開発」となる。AI モジュールの開発では、AI 特有のアジャイル的な要素を考慮している。AI モジュールの開発の中では、AI モジュールの設計に向けて、まず「AI 要求分析」を行い、次に「データセット設計・収集」と「学習モデル開発」でデータセットの質と量、および学習モデルとその学習方法のそれぞれについて検討を行う。その結果のデータセットや学習モデルに関しては、PoC 検証を含む「AI 検証」を行い、要求機能および性能を実現できない場合は、データセットや学習モデルの検討を繰り返し行い、機能実現および性能達成を目指す。

なお、「AI モジュールの開発」では、PoC 初期段階／製品開発段階／保守・運用段階ごとの目標達成状況に応じて、プロセスの繰り返し範囲は異なる。

上記の 3 つのリスク低減方策の検討の後、「システム検証」において、当初想定していたシステム全体でのリスク低減の達成を確認する。ここで、想定したリスク低減が達成できない場合は許容できるリスクになるまで繰り返し「システム RA」に戻り、リスク低減方策の再検討を行う。

以上のような開発プロセスに従うことで、上流から一貫した開発を行うことができ、目的の性能（利用時品質 : 安全性・リスク回避性）を達成するために各プロセスで何をするべきかが明確になる。さらに、検討事項の抜け漏れが無くなり、首尾一貫した開発が行えるようになる。

なお、この開発プロセスの左部分については、基本的には従来の機能安全の開発プロセスを踏襲するものであり、これに右半分の AI 特有の AI モジュールの開発プロセスが加わる形となっている。そのため、従来のシステム安全設計と容易に融合できるものとなっており、従来の機器に AI モジュールを加える場合でも、従来の機能安全の設計の考え方や開発プロセスは、そのまま流用が可能となっている。

また図 2 では、機械学習品質マネジメントガイドラインで規定された AI の品質特性である「内部品質」と、それぞれの開発プロセスとの対応づけを示しており、どのプロセスでどの内部品質を検討するべきかを示している。ただし、開発フローについては、代表例を示したものであり、記載した以外の繰り返し範囲も想定される。

6.3 品質アセスメントシート

前述した開発プロセスの構築と併せて、これらのプロセス実施を支援する品質アセスメントシートの開発も行った。このアセスメントシートでは、各プロセスに対応したシートを用意している。それらのシートは、それぞれプロセスに対応する機械学習品質マネジメントガイドラインの内部品質を評価可能としたものであり、そのシートの項目を記載していく中で自動的に内部品質の検討・評価が可能なものとなっている。この開発プロセスでは、図 3 に示すようにそれぞれの各プロセスの入力と出力を規定しており、前のプロセスの出力が後ろのプロセスの入力になることで首尾一貫したプロセスとなっている。作成した品質アセスメントシートでは、これらの入力から出力を作成し、プロセス間の記載の対応を示すト

レーサビリティが確保できるシートとなっている。

以下では、各シート内容を説明する。PoC 初期段階、PoC 最終段階／商品開発開始段階、商品開発完了段階、運用段階といった各段階で、シートに記載可能な内容およびその詳細さは異なると考えられるため、各段階で記載が必要な項目を色分けで示した。

なお、「機能安全開発」については、従来の機能安全の開発プロセスとの違いはないため、新たにアセスメントシートの作成はしていない。このプロセスについては、必要に応じて、AI を用いていない従来の開発プロセスで使用しているものを流用すればよい。

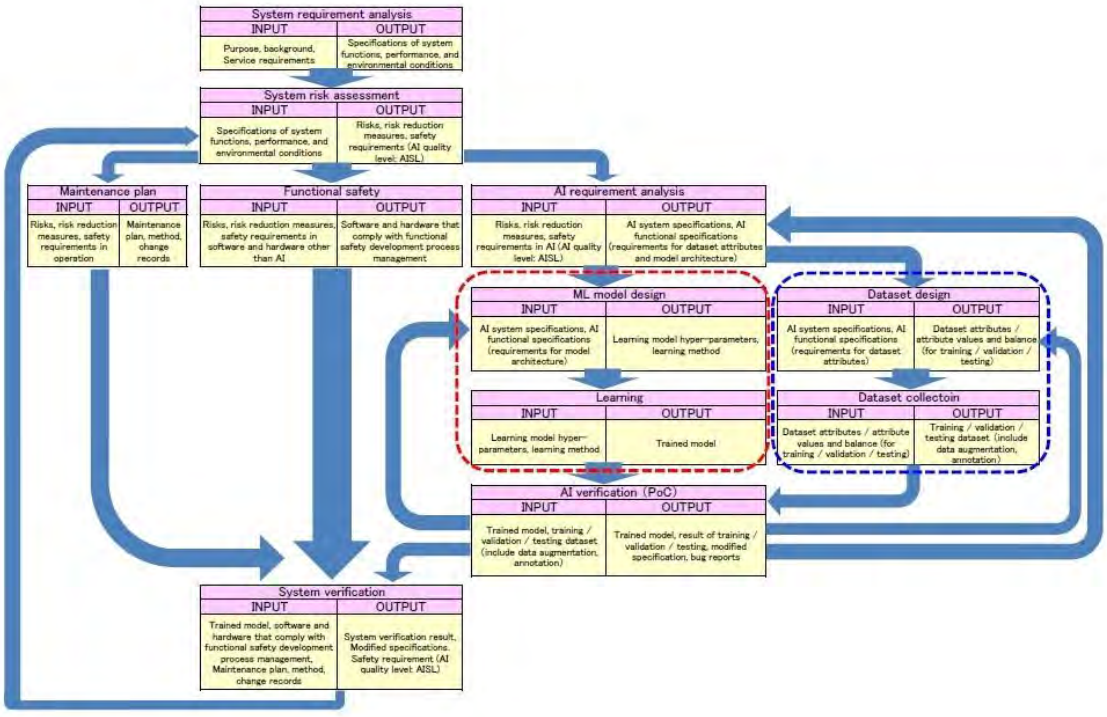


図 3 各プロセスの入出力

6.3.1 システム要求分析票

【対象プロセス】システム要求分析
【入力】使用目的、サービス要求事項
【出力】システム機能（ユースケース）、性能、環境条件、使用制限、システム構成

システム要求分析のプロセスでは、使用目的、使用方法、環境、などの使用シーンに相当するユースケースからサービスの要求事項を抽出し、システムの仕様を決定する。「システム要求分析票」には、本プロセスにおいて使用に関連する内容（ユースケース）と制約事項を書き出すことで整理を行い、システムの機能や性能、環境条件、システムの構成に落とし込む。



図 4 システム要求分析票

帳票のフォーマットとして、図 4 を例示しているが、一般的な要求分析と同様に、要求事項を整理できるものであれば、特に本例のフォーマットに限定するものではなく、整理しやすい形式であればどのようなものでも良い。

6.3.2 システム・リスクアセスメント票

【対象プロセス】システム RA（リスクアセスメント）

【入力】システム機能、性能、使用制限、システム構成

【出力】リスク見積り、リスク低減方策、安全度要求レベル

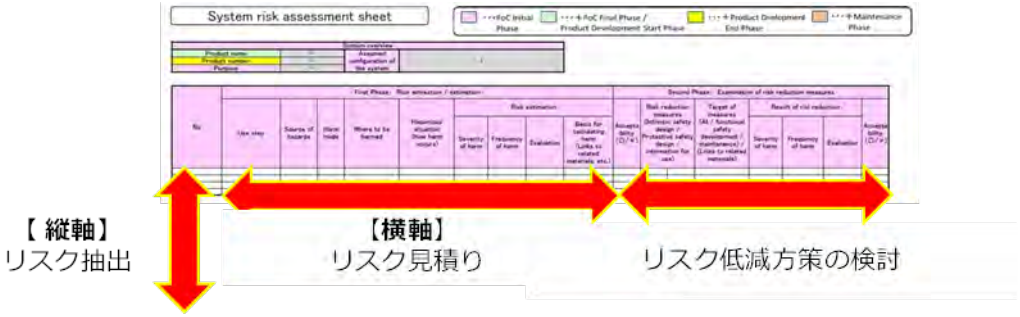
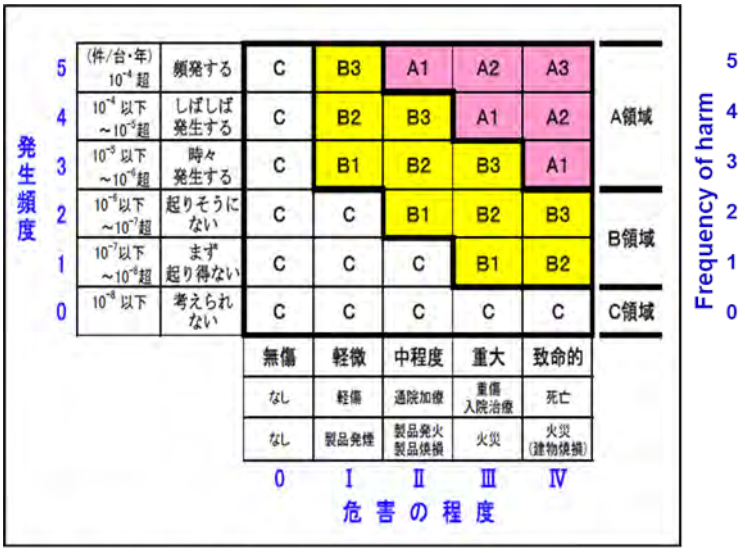


図 5 システム・リスクアセスメント票



*本リスクマップは、リスク定義の一例であり，
この形式に限定されるものではない。
* "Applying the R-Map Method to Product Safety and Risk Management, Japan", は、
リスク管理方法の一つとして、ISO13077:2013(en)から参照されている。

図 6 リスクマップの例

システム RA のプロセスでは、システム要求分析をもとに構想設計されたシステムの RA を行う。この「システム・リスクアセスメント票」では、本プロセスにおいてシステムのリスクの洗い出しを行い、それぞれのリスクの見積もり、低減方法、低減目標（安全度要求レベル）を決定する。決定したリスク低減方策については、内容に応じて、AI モジュールに関するもの、ハードウェア・ソフトウェア設計に関するもの、保守・運用に関するもの、の 3 つの観点で、それぞれ以降のプロセスに引き継いでいく。

図 5 に示した帳票のフォーマットは、一般的に使用されるリスクアセスメント帳票と同様であり、リスクの抽出、リスクの見積り、リスク低減方策の検討を記録する。リスクの見

積り手法については、図 6 の例に記載したリスクマップを用いる方法などがあるが、それに限定するものではなく、方針を明記し、第 3 者にも定量的に説明できるものであればよい。

なお、従来との違いは、リスク低減方策に AI によるリスク低減が加わり、安全度の要求レベルとして、IEC61508 機能安全の SIL や ISO13849-1 の PL(Performance level)に相当する AISL が、機械学習品質マネジメントガイドラインに基づき指定されることである。

6.3.3 AI 要求分析票

【対象プロセス】AI 要求分析
【入力】AI に関するリスク見積り、リスク低減方策、安全度要求レベル
【出力】AI システム仕様、データセット、学習モデルへの要求仕様

AI 要求分析のプロセスでは、システム RA により振り分けられた AI によるリスク低減方策、安全度要求レベルに対して、AI 利用システムの仕様、データセット、学習モデルへの要求仕様を設定する。

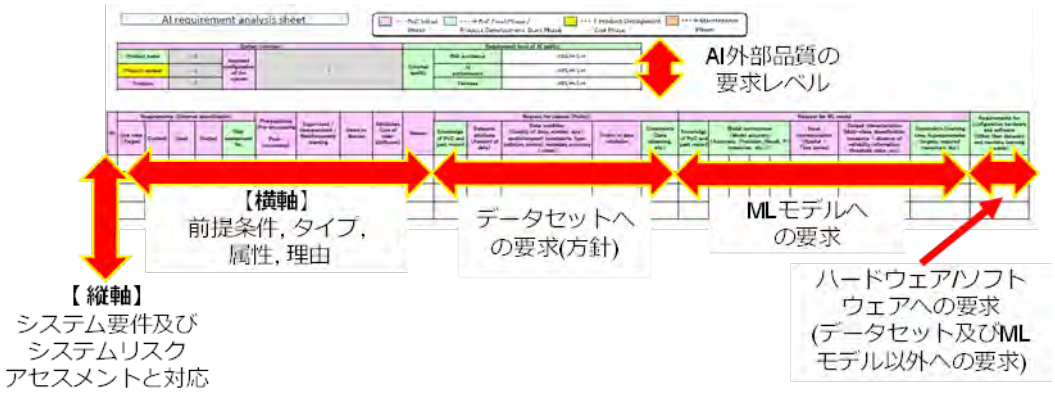


図 7 AI 要求評価票

帳票のフォーマットとしては、図 7 に示したように、一般的な要求分析に加えて、AI 特有のデータセットおよび学習モデルに関する方針についての要求事項が必要となる。具体的には、「AI 要求分析票」では、AI 利用システムにおける AI モジュールに関して、目的に適した AI の種類を選択する。教師あり学習の場合には、その学習対象の特徴として着目する主な属性を洗い出し、データセット、学習モデル、およびそれ以外のハードウェアやソフトウェアへの要求事項を、その前提条件および理由とともに列挙、明確化し、記録を支援する。開発段階において、記載可能な範囲は変わってくることが想定されるため、例えば以下のような目安で記載する。

PoC 初期段階では、システム要件を踏まえて、AI に期待する基本的な役割、必要と考えられる前提条件（事前処理、事後処理）、適用候補となる学習手法（AI のタイプ）を検討

し、学習に用いるデータセットの収集・作成についての方針、学習モデルの入出力特性を検討して記載する。

PoC 最終段階/製品開発開始段階では、PoC 初期段階での検討内容の見直しに加え、システム RA 結果との対応付け、PoC および過去の記録による知見、制約事項、学習モデルの精度要求を記載する。

なお、組織としての AI 開発標準を設け、開発計画段階において、対象となる AI 利用システムの開発プロジェクト向けにテーラリングする場合には、計画書で開発段階ごとの記載範囲ルールを定め、帳票をカスタマイズして利用するなどの使い方も想定範囲内とする。

6.3.4 データセット・アセスメント票

【対象プロセス】データセット設計・収集
【入力】AI システム仕様、データセットへの要求仕様
【出力】データセットの設定（データセットの属性、属性毎のデータ数）

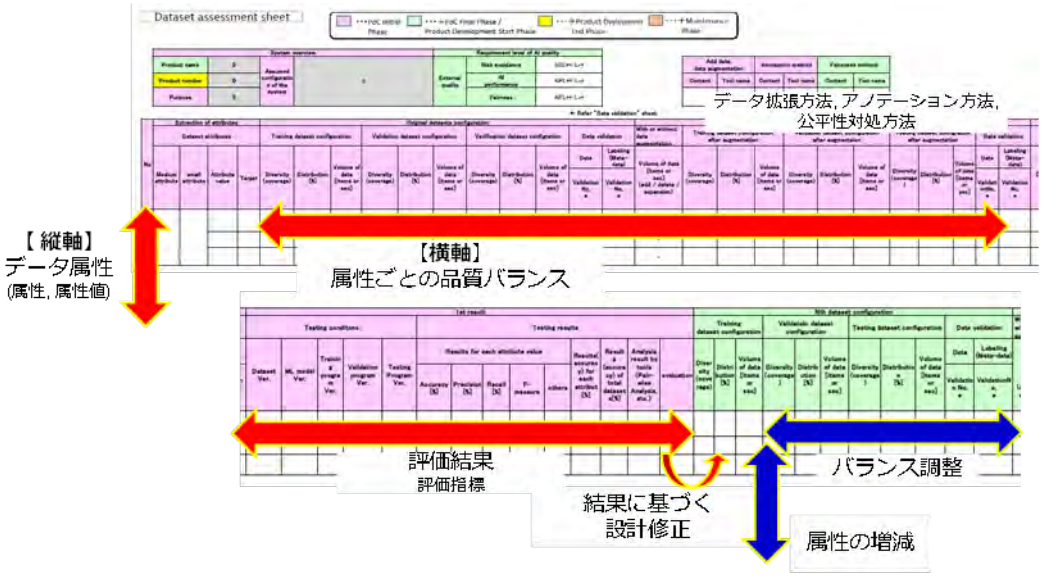


図 8 データセット・アセスメント票

データセット設計・収集のプロセスでは、AI 要求分析により設定された AI 利用システム仕様およびデータセットへの要求仕様に対して、データセットの属性、および属性毎のデータ数量を記録する。図 8 に示すように、「データセット・アセスメント票」では、データセットの属性を列挙し、その属性毎のデータ数量の初期値を記載する。次に、AI 要求分析時に規定したポリシーに従い、データセットを検証し、必要に応じて、データセットの拡張、アノテーションの調整、公平性への対応を行う。そして、次の AI 検証(PoC)プロセスに移り、そのデータセットでの AI 機能、性能の検証を行う。そしてその結果を品質アセスメントシートに記載し、目標の機能、性能に到達しない場合は、その検証結果をもとに属性の増減や属性毎のデータ数のバランスを調整、「データの妥当性」も確認して、再度検証を行う。

これを目標達成まで繰り返す。

帳票のフォーマットとしては、図 8 に示すように縦軸に属性を列挙し、それぞれの属性に対し、横軸で属性毎のデータ数、そのデータセットでの検証結果を記述するようになっており、これを 1 セットとして繰り返し分が横軸に並ぶ形式となっている。ガイドラインの内部品質に関して、縦軸は「データセットの被覆性」、横軸は「データセットの均一性」に関連しており、この品質アセスメントシートでデータセットの内容を確認しながら、これらの内部品質を調整していく。

「データの妥当性」については、図 9 に示すように、データに対しては、縦軸にはデータ取得のための条件、横軸にはデータの出所（新規/加工/流用）をはじめ、時間的妥当性、空間的妥当性、外れ値除去、汚染可能性、汚染対処方法、検査方法、確認結果、データの使用可否判断など、妥当性確認内容を記載する。また、ラベリング情報などのメタデータについても、データと同様の妥当性確認内容に加え、AI 要求分析において設計したラベリングのポリシーに準拠しているか、などの妥当性確認を行う。

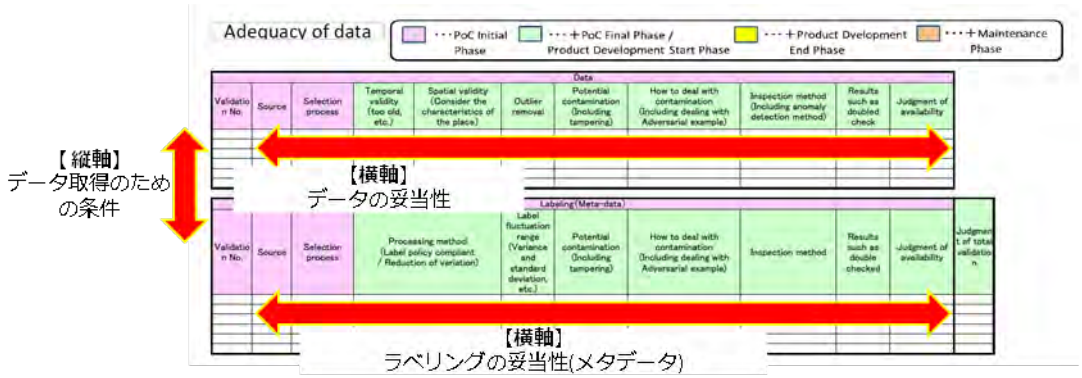


図 9 データの妥当性

6.3.5 学習モデル・アセスメント票

【対象プロセス】機械学習モデル開発

【入力】AI 利用システム仕様、学習モデルへの要求仕様

【出力】学習モデルの設定（ハイパーパラメータ、学習方法含む）

機械学習モデル開発のプロセスでは、AI 要求分析により設定された AI 利用システム仕様および学習モデルへの要求仕様に対して、学習モデルのハイパーパラメータおよび学習方法の記録を行う。データセットの設計・収集のプロセスと同様に、「機械学習モデル・アセスメント票」において、ハイパーパラメータや学習方法を記録後、その学習モデルでの AI 機能、性能の検証を行う。そしてその結果を品質アセスメントシートに記載し、目標の機能、性能に到達しない場合は、その検証結果を基にハイパーパラメータや学習方法を調整して、目標達成まで繰返し検証を行う。

帳票のフォーマットとしては、縦軸にハイパーパラメータを列挙し、横軸で学習方法や手順、その学習モデルでの検証結果を記述するようになっており、これを 1 セットとして繰り返し分が横軸に並ぶ形式となっている。また、学習モデル自体を変更、複数の機械学習モデルから比較検討して選定する場合には、過去の学習モデルに対する検討結果は残した状態で、新たな学習モデル用のシートを作成し、新しい学習モデルでの内容を記入する。

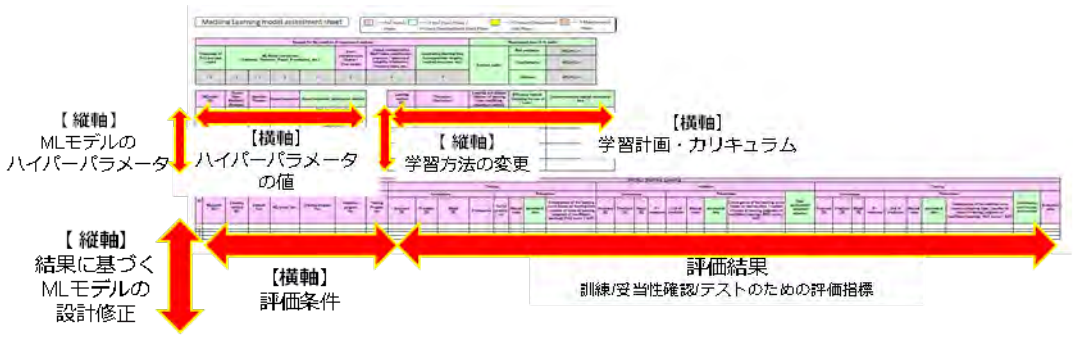


図 10 機械学習モデル・アセスメント票

6.3.6 保守計画アセスメント票

【対象プロセス】保守計画
【入力】保守・運用に関するリスク、リスク低減方法、安全度要求レベル
【出力】保守（運用）計画・実施結果

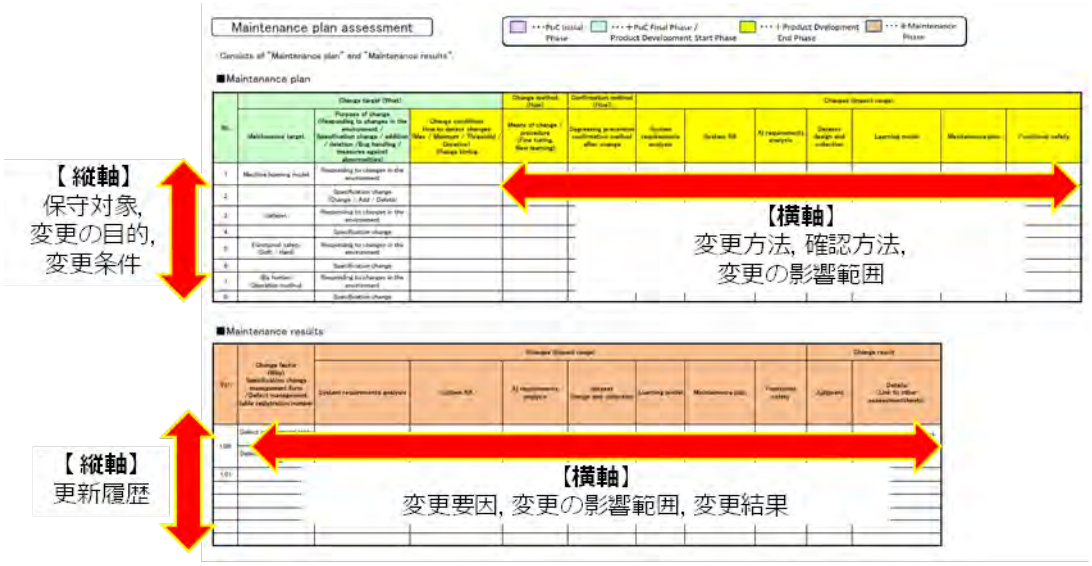


図 11 保守計画・実績アセスメント票

保守計画プロセスでは、システム RA により設定された保全・運用に関するリスク、リスク低減方法、安全度要求レベルに対して、保守や運用の計画を行い、これを実行する。AI

モジュールのデータセットや学習モデルの変更を計画する場合は、逐次学習や環境変化への対応などに合わせて、どのように変化させていくのかを計画する。また、実施については、計画されたものに加え、計画外の実施も管理対象とし、変更結果について履歴を残す。

帳票のフォーマットについては、計画と実績の 2 つの表に分け、計画においては運用および保守に関する計画を記録する。計画では、変更の対象、目的、条件、手段、確認方法、変更の影響範囲を記録する。実績については、計画外も含めて、変更履歴が判るように記録し、変更の影響範囲と変更の結果も記載する。また、この実績と併せてデータセットや学習モデルの変更、調整については、その状況をデータセット・アセスメントシートや学習モデル・アセスメントシートに記載し、デグレーションがないことを確認できるようにしている。

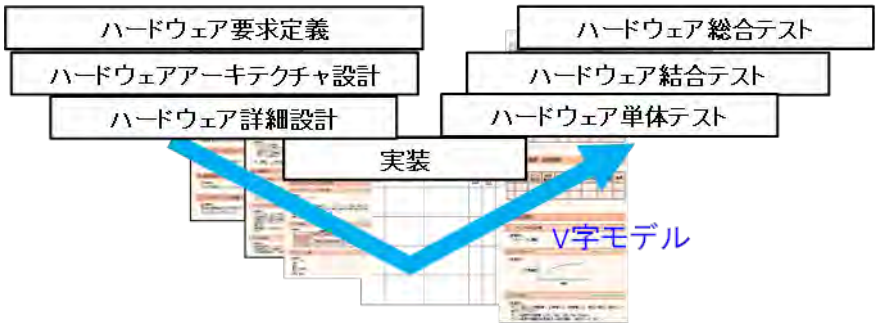
6.3.7 機能安全開発

【対象プロセス】機能安全開発（その他のハードウェア、ソフトウェアの開発）
【入力】システムのハードウェア、ソフトウェアに関するリスク、リスク低減方策、安全度要求レベル
【出力】機能安全対応のハードウェア・ソフトウェア

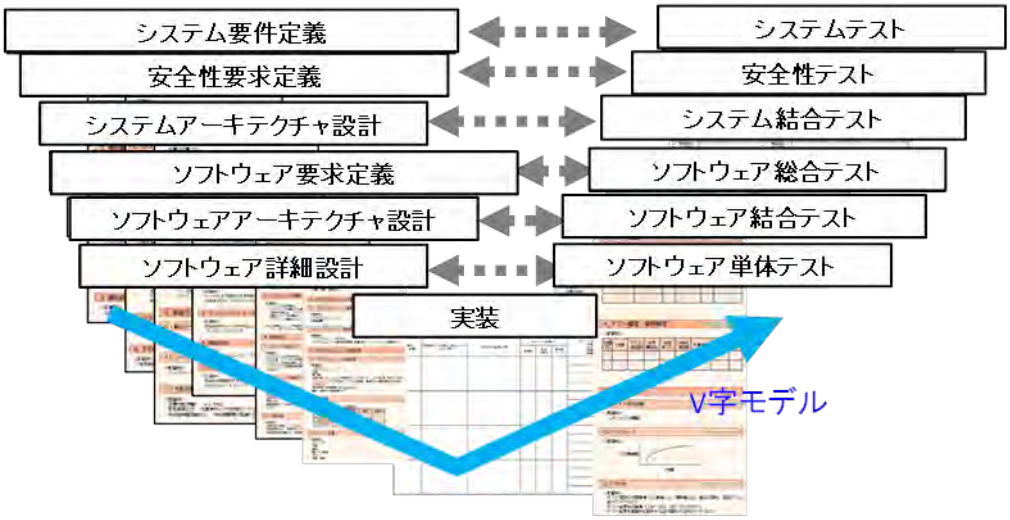
機能安全開発プロセスでは、システム RA により設定されたシステムのハードウェア・ソフトウェアに関するリスク、リスク低減方法、安全度要求レベルに対して、機能安全に対応したハードウェア・ソフトウェア開発を行うものであり、AI モジュールの処理性能以外に一般的な機能安全開発と違いはない。参考までに図 12 に一般的に機能安全開発で用いられる V 字モデル開発を示す。

なお、V 字モデルは各開発プロセスの位置づけや対応関係を示すものではあるが、開発順序を示してしているわけではない点に注意が必要である（即ち、ウォーターフォール型だけではなく、スパイラル型など繰り返し型の開発も含んでいる）。

本プロセスでの帳票は図 12 に示される各種設計書やテスト仕様書・結果など、従来の機能安全開発に用いるドキュメントを用いれば良く、特に新たな帳票は作成していない。



(a) ハードウェアの開発



(b) システムおよびソフトウェアの開発

図 12 機能安全の V 字モデル開発

6.4 開発プロセスおよびアセスメントシートの適用例

前述の品質アセスメントシートを様々なプロジェクトに適用することで、「開発プロセス」とそれを支援する「品質アセスメントシート」の実証を行った。それらでは、図 13 に示すように、品質アセスメントシートを各開発プロセスで利用した。

本章では、インテリジェント車椅子の開発への適用を行った場合について、特に AI 特有の部分である「データセットの設計・収集」プロセスにフォーカスをして、「データセット・アセスメントシート」を記載する手順の具体例を示す。

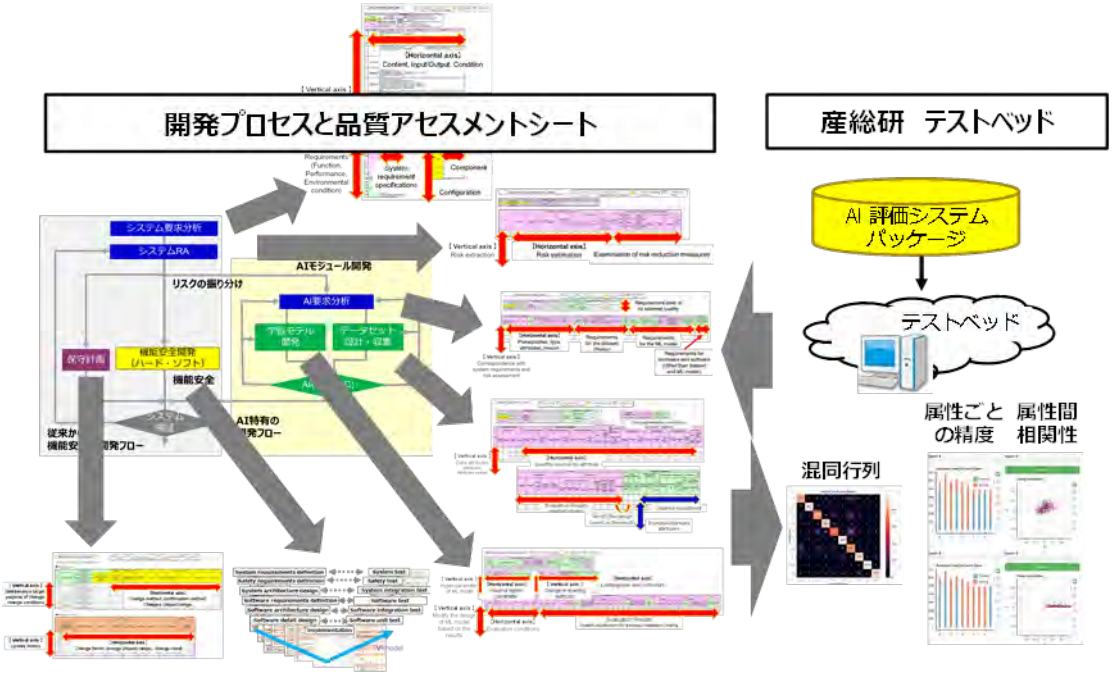


図 13 開発プロセスとそれを支援する品質アセスメントシート

6.4.1 適用される AI の内容

本適用例では、画像センサと AI による画像認識機能を持ち、周囲の人を認識して衝突を回避するという安全制御を自律走行する車椅子型のロボット（インテリジェント車椅子: 人を乗せていない場合は、自律移動搬送車として機能）の開発に適用した場合について、部分的な実証例を示す。適用先としたインテリジェント車椅子の外観を図 14 に示す。

インテリジェント車椅子では、表 A-1 のような既存の AI のデータセットおよび学習モデルを使用して人検出を行うものとする



図 14 ターゲットとしたインテリジェント車椅子（2019 年試作品）

インテリジェント車椅子では、表 1 のような既存の AI のデータセットおよび学習モデ

ルを使用して人検出を行うものとする。

表 1 インテリジェント車椅子での AI の構成

Item	Content
Dataset	COCO dataset（2014）
ML model	YOLOv3（Darknet 53）。 Trained machine learning model (yolov3.weight)

データセットとして使用した COCO dataset は，画像データとそのメタデータとなるラベリング情報（画像に含まれる物体の種別とその画像内での対象領域を示す枠である Bounding Box の 2 次元位置を表す）を含む。また，学習モデルのアーキテクチャである YOLOv3 は，80 種類のカテゴリー（クラス）で物体検知する機能を持ち，インテリジェント車椅子で検出する「人」はその検出対象物体の一つである。本章のインテリジェント車椅子では、これらのデータセット、学習モデルのアーキテクチャによって予め訓練された訓練済機械学習モデルを実際の人検出に用いて、これらを初期値としてデータセットおよび学習モデルを構築していくものとした。

6.4.2 システム要求分析

本章では、データセットにフォーカスをした適用を進め、まず、図 4 に示す「システム要求分析票」を用いて、全体システムの要求分析を行う。インテリジェント車椅子のシステムの機能要件、および非機能要件である性能・環境条件について記載した。特に、認識の判断データとなるカメラ画像の取得に影響するロボット自体の動き、明るさ等の環境条件、さらに画像に写る可能性のある障害物や人の特性などもここで明らかにする。また、リスクの振り分けにも関わるシステムの運用についても言及し、最終的に AI に求める要件を可視化するような分析を行っている。この分析では、ユーザ側の要望・シーンに対して、AI の設計開発者だけでなく、システムの設計者など技術担当者のシステム化において検討すべき技術ポイントを洗い出し、これらの観点を意識しながら分析、システムの構想設計を行うことで以降のプロセスにスムーズに移行できる。

なお、本節に示すのは品質アセスメントシートの記載例であるため、一般的に使われるユースケースから絞り込んだシーンのシナリオに対象を限定している。

6.4.3 システムリスクアセスメント

本プロセスでは、システム要求分析をもとに構想したシステムに対し、図 5 に示す「システム・リスクアセスメント票」を用いて、リスクの抽出とリスクへの対応方法を検討している。ここでは、一般的なリスクアセスメントと同様の検討を行うため、詳細についてはここでは省略するが、ポイントは AI で対処すべきリスクおよびそのレベルを十分に検討しておくことが必要となる。十分な考慮無しに AI にリスク低減方策を振り分けるべきではなく、

ISO/IEC Guide 51 and ISO 12100 に記載されている安全設計の基本となる 3 ステップメソッドに則り、本質安全設計から制御による安全設計へ、などと適切にリスク低減を図る。また、設計として最後に AI が担うリスクは許容できる範囲まで小さくなるように、AI にしかできないことでのリスク低減に制限をしている。これは、現時点の技術では AI によるリスク回避性は、まだ信頼できるレベルではないためであり、最大での AISL0.2 と実際の機能安全の SIL1 以下のレベルとしている。

なお、このリスクアセスメントもシステム要求分析と同様に絞り込んだシーン・シナリオに対象を限定して実施している。

6.4.4 AI 要求分析

システム・リスクアセスメントの結果に基づき AI へのリスク低減方策の内容に関して、今度は AI への要求分析を図 7 に示す「AI 要求分析票」を用いて行った。ここでは、AI で認識するべき「属性」を明らかにし、取得画像から認識対象の「人」とその他の対象物を分類する。まず、システム要求分析でのシーンと結び付けて、要求内容に基づき、認識の対象物、その対象物に対する関連属性を洗い出している。そして、それらの属性のリスクへの影響を考慮しながら、AI の属性として採用するかの検討、絞り込み、またはさらなる分類を行う。ここでのポイントは、可能な限りの属性を洗い出しておくことであり、もし最初に候補に入れなくても、実際の開発の中でうまく認識ができない場合は、それらの項目を属性として再考できるものとする。

なお、本適用ではデータセットを例に検討を行った。AI 要求分析では、抽出すべきデータの属性について、まずはデータ収集・設計の方針および優先順位を決めておき、以降に続くデータセット・アセスメントにおいてデータの内部品質の詳細を確認する。

6.4.5 データセットの設計と収集

データセット設計・収集のプロセスでは、AI 要求分析で抽出した属性をもとに図 8 に示す「データセット・アセスメント票」を用いてデータセットの設計・収集を行った。ただし、本章ではデータセットがすでに与えられているものを用いるため、「AI 要求分析」で抽出した属性によって、データセット「COCO dataset (2014)」を分析している。

ここでは、まず属性について AI 要求分析の結果をもとに図 15 のような属性値を記入し、データセットを分析した。図 16 はその分析の部分的な例であり、COCO dataset の訓練データ 82081 件分に対して、属性（カテゴリー）ごとのデータ数を降順に並べ、各属性内の属性値（例「明るさ」）のデータ数(図 16(a))を示し、その属性順に属性内の属性値の相対分布(図 16(b))を示したものである。

ただし、図 15 で示した小属性は、中属性を階層的に分類する際の属性の観点の一例として挙げたものであり、中属性に応じて小属性も変わるものである。例えば、中属性が「人」の場合には、小属性としては「明るさ」以外にも「年齢」「姿勢」「身体部位」なども一種として考えられる。中属性が「自転車」の場合には、小属性として「種類」、「色」なども考えられる。

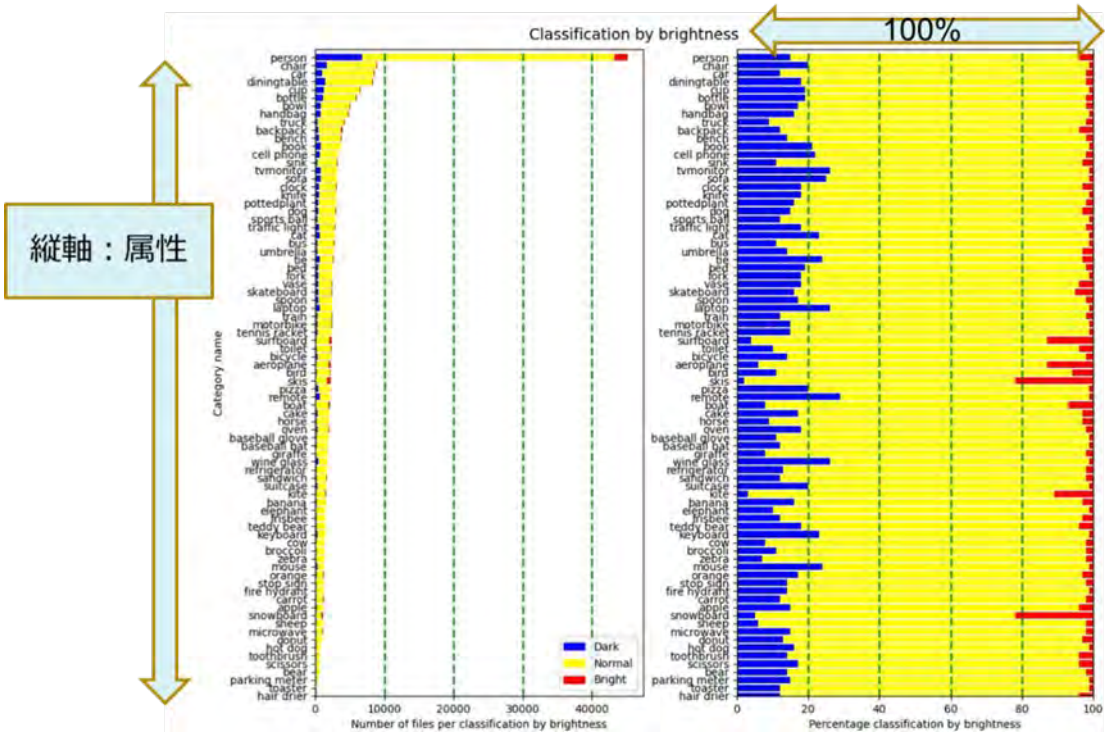
No	属性の抽出				訓練用データセットの構成			
	データセット属性							
	中属性	小属性	属性値	対象	被覆性	分布[%]	データ数または量[件 or sec]	
	人	明るさ		✔	6795	15	45174	
				✔	36445	81		
				✔	1934	4		
	自転車	明るさ	明るい		36	2	2287	
			普通		1939	85		
			暗い		312	14		
	車	明るさ	明るい		144	2	8606	
			普通		7403	86		
			暗い		1059	12		
	バイク	明るさ	明るい		26	1	2442	
			普通		2039	83		
			暗い		377	15		

図 15 データセットにおけるデータ属性と属性値の例

本プロセスにおいて重要な点の一つは、品質アセスメントシートの利用を通して、属性について様々な観点で考え、目的のために必要となるデータセットの特徴を洗い出し、それらの取捨選択を検討することである。また、その検討結果を記録することで、データセットに関する２つの内部品質（すなわち、属性に対するデータの抜け漏れがないかの観点の「データセットの被覆性」およびデータ分布のバランスを確認する観点の「データセットの均一性」）を定量的に可視化し、エビデンスの材料とすることができる。

さらに、結果的に、繰り返し行うテスト結果と照らし合わせて、不具合の分析および改善ポイントの抽出など、その後の品質改善活動にも役立てることができる。

一方、データに関する内部品質「データの妥当性」を確認しておくことで、ラベリング誤りなど、機械学習モデルの検証時に現れる不具合リスクをデータ設計・収集時に低減することも可能となる。



(a) 絶対的なデータ分布　　(b) 相対的なデータ分布

図 16 データセットの属性毎の属性値（明るさ）に関するデータ量の分布
（YOLOv3 用 COCO datasets）

6.5 まとめ

本章では、AI 品質管理のために検討した開発プロセス、および開発プロセスの推進を支援するために開発した品質アセスメントシートの使用方法を紹介した。

なお、帳票については、本資料で例示したフォーマットを固定的なものとして考えるのではなく、知見や新たな技術の反映を通して、今後も更新していくものとする。

7 自動運転車

自動運転車のための物体検知と場面識別のための AI モジュール。

7.1 概要

ある AI 製品/サービス開発チームは、ある自動車メーカーから自動運転車用 AI モジュール構築のためのビジネス要件を受け取った。開発チームが受け取ったビジネス要件文書（BRD）は、7.2 節に示す。次に 7.3 節において、受け取った BRD に基づいて、問題の予備的な分析を行う。次に、Proof-Of-Concept フェーズを実施し、それからプロトタイプの開発を計画する。最後に、MLQM ガイドラインの内部品質特性をプロセス全体に渡って検証することにより製品品質を評価する方法を示す。

7.2 節では、架空の BRD を提示して、BRD や同様の文書を作成する際に、MLQM ガイドラインをどのように取り入れるか、また、取り入れることでプロセス全体にどのような利点があるかを例示する。

それ以降の節では、製品・サービスの開発プロセス全体について、特に各フェーズにおける品質基準の評価に焦点をあてて解説する。これらの節は、企画から開発までの品質特性の評価と管理に関する MLQM ガイドラインの実施方法を理解するための鍵となる。

- 7.3 節は、前項の BRD に基づいて、開発者の視点から製品の主要な技術仕様を特定したものである。安全性や性能に関する仕様も含む。
- 7.4 節は、PoC フェーズについて述べる。初めに、既存のデータセットや課題解決手法に関する調査結果を示す。次に、データの予備的な分析と予備的なトレーニングの結果を示し、最後に PoC フェーズから得た洞察を論じる。
- 7.5 節は、MLQM ガイドラインの内部品質特性軸の評価結果を示しながら、開発段階について検討した事項を述べる。
- 7.6 節は、この例のための用語集を示す。

7.2 ビジネス要件

7.2.1 製品名

自動運転車に使用される物体検知や場面識別を行う AI モジュール。

7.2.2 ユースケース

この AI モジュールは、以下のユースケースを想定している [1]。

(a) ケース 1：様々な運転シナリオで通常遭遇する物体の識別と位置特定

(b) ケース 2：半自動運転車の標準的な環境条件（天気、道路種別、信号など）の認識

7.2.3 背景

ある自動車メーカーが、自社の自動車に自動運転機能を搭載したいと考えている。この車両には、前方に単眼カメラが取り付けられており、周囲の環境を連続的に撮影することができる。AI モジュールに期待されているのは、ビデオストリームから抽出したフレームを受信して、運転中に頻繁に遭遇する物体を識別し、位置を特定することである。また、天気、交通、道路状況などの環境条件も評価する必要がある。

SAE（Society of Automotive Engineers）International 規格 J3016 “Levels of Driving Automation” によると、この自動運転の目標レベルは SAE レベル 1 である。これは、自動運転車がアダプティブクルーズコントロール（定速走行・車間距離制御）機能の運転支援を行うことを意味する。すなわち、次の車との安全な車間距離を保ちつつ、人間の監視のもとでハンドルを操作する。

この段階では、AI モジュールはドライバーの運転体験を向上させるためにのみ使用される。特定された環境条件の組合せが、あらかじめ定義された安全上重要なシナリオ（信号が青なのに横断歩道に歩行者がいるなど）と一致した場合、アラームシステムが警告を出し、ドライバーはそれに従って行動し、事故を回避することができる。

メーカー側は、公開されている動画を使って AI を開発することで、何が実現できるかをまず確認することで合意した。

7.2.4 目的・目標

運転シナリオの常時監視による運転体験の向上。

安全を最重視すべき場面で警告を発することによる、事故確率の低減。

完全自動運転への第一歩としての、一般車両への自動運転機能の搭載開始。

7.2.5 製品のステークホルダー

このガイドで考慮する本製品のステークホルダーは以下の通りである。

---

[1] ユースケース記述としては、目的や用途だけでなく、ターゲットに対する入力／処理／出力を明確にして、どのようなやり取りを行うかを明確にすることが望ましい。

- 自動車メーカー
- お客様（ドライバー／公共交通事業者／自家用車所有者）

7.2.6 ステークホルダーの初期要求

- 自動車メーカーやお客様からは、ユーザーや車両の安全性を最優先して開発することが求められている。
- 自動車メーカーからは、容易に製品に搭載できるように、このモジュールは現在のハードウェア要件に適合させてほしいとの意向が示された。
- 安全性だけでなく、お客様は AI モジュールのインターフェースが使いやすく、低速域でも高速域でも最大限の機能を発揮することを希望している。
- この製品の最大の関心事は、ユーザーと車両の安全性である。AI モジュールは、安全上重大な事象をリアルタイムで効果的に検知し、命に関わる事態の回避を支援できるべきである。
- このモジュールは、通常の環境（晴天など）だけでなく、あまり遭遇しない環境（雨天や雪天など）でも物体を識別できることが推奨されている。もし、遭遇率の低い運転状況でモジュールが同様の性能を発揮できなければ、事故がより発生しやすくなる恐れがある。

7.2.7 ビジネス要件の詳細

開発する製品のビジネス要件については、以下で詳しく説明する。

機能要件

最終的な AI モジュールの機能要件は以下の通りである。

- 天気や道路状況その他の特徴に基づいて、現在の運転状況を識別する。
- 気候条件、交通条件、道路条件、時間帯、照明条件など、あらゆる条件下で、歩行者などの人、車、信号、標識、バス、バイクなど、運転中に日常的に遭遇する物体を検出する。

補足：車両全体の自動化レベルは SAE [2](図 17) レベル 1 になる。AI モジュールの物体検知機能と場面識別機能は、これらを車両が使うことによって、アダプティブクルーズコントロールを行うに足る安全な走行状況かどうかを確認し、また、運転者が直ちに注意を払うべき安全上重大なシナリオが発生していないかどうかを確認する。

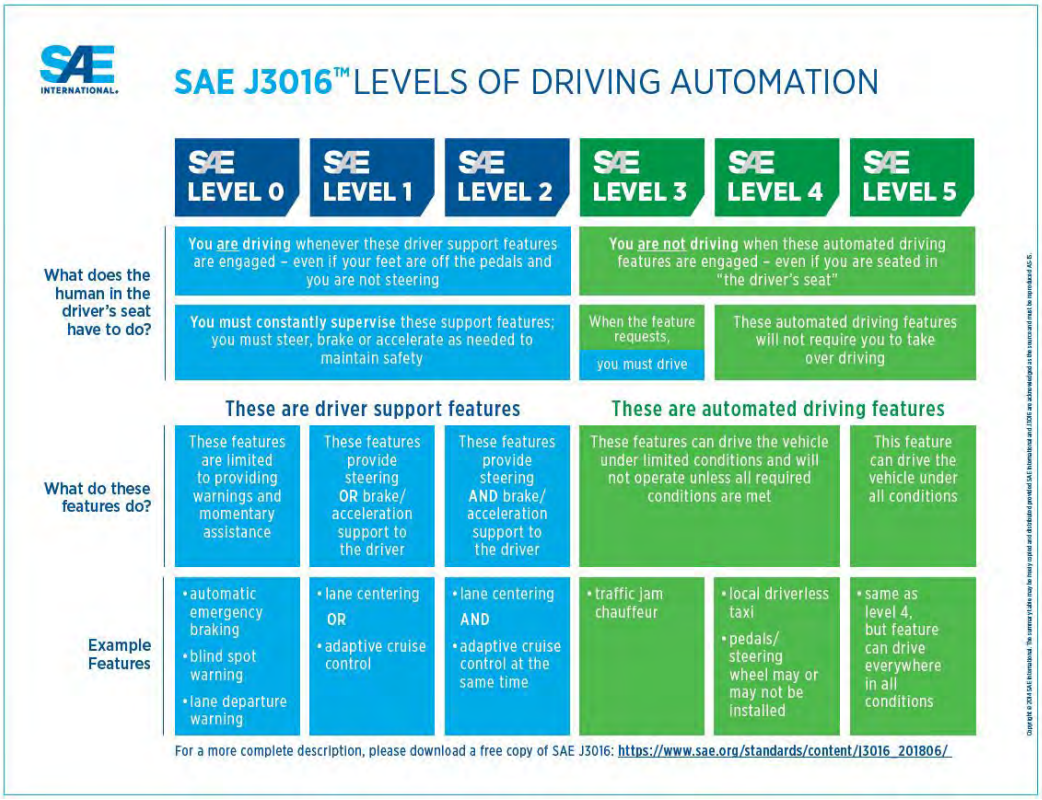


図 17 自動運転の SAE レベル

非機能要件

AI モデルの非機能要件は以下の通りである。

- AI モジュールは、FPS（毎秒処理フレーム数）で与えられる最低限の速度での処理において、必要な精度を維持する必要がある。
- AI モジュールは、自動運転車が走行する米国の都市部や郊外の交通ルールや状況に最も適合している必要がある。
- AI モジュールは、すくなくとも比較的低い走行速度では十分に動作すること。
- 物体検知と場面識別の機能は、稀な運転状況においても頑健である必要がある。
- AI モジュールは、分かりやすいインターフェースがあって使いやすいこと。

想定

- 一般に入手できる公開走行映像には、運転状況、天気、道路状況などの様々な組合せの例が存在する。
- 人間が画像中に存在するすべての物体を認識するのに十分適切な照明がある。

- AI モジュールが警告を発するべき、事故が起こりやすい／安全上重大な状況は、一般に入手できる公開走行映像の中に存在する。

依存事項

- **動作中**　単眼カメラで車両前方の映像を連続撮影し、同時に別のシステムで映像からフレームを抽出し、入力画像として AI モジュールに送る。これらのシステムの誤動作は、AI モジュールの故障につながる。これらのシステムの正確な同期を定期的にチェックし、維持する必要がある。カメラは汚れや水滴が付きにくい形で車両内に設置されており、定期的なチェックにより良好な視界を維持する必要がある。

制約事項

- AI モジュールを構築する際の画像の主な収集源として、一般に入手できる公開走行映像データセットを使う必要がある。

考慮しない事項

- **機能**　この AI モジュールの機能は、物体検知と場面識別に限定される。安全上重大な場面での警告発出や、安全な場面でのアダプティブクルーズコントロールなど、その他の自律的な機能は、車両の他の個別モジュールが担当する予定である。これらのモジュールは、ここで開発する AI モジュールから周辺環境に関する出力を受け取るのみである。
- **走行地域**　前述の通り、AI モジュールの学習には特定地域の走行映像を用いる。交通ルールや運転状況は世界各地で異なることがある。また山中や海浜など道路整備がされていない地域も全く状況が異なる。交通ルールや運転状況が前述の地域と大きく異なる地域は、本 AI モジュールの対象外とする。本 AI モジュールを搭載した車両を他の地域に配備する場合は、事前に再学習を行い、適切に調整することを推奨する。
- **経時的変化**　交通ルールや運転状況は数年から数十年で大きく変わる可能性がある。交通ルールや運転状況が AI モジュールの学習に使う走行映像とは大きく変わった場合、それ以降の本 AI モジュールの使用は想定しない。
- **速度の上限**　高速走行ではこのモジュールの性能が不足することがある。そのため、高速で移動する車両に対する支援を目的としたモジュールの使用は推奨されない [2]。

リスクと安全に関する懸念

- AI 製品が故障した場合、人命を脅かす事故や経済的損失を引き起こす事故が発生する可能性がある。そのようなリスクを軽減するために、開発プロセスを通じて注意と対策を講じる必要がある。例えば、ある条件下（雨天・霧天など）で視界が悪いと、AI モデルの性能に悪影響を及ぼし、モジュールの誤動作の原因となること

[2] 実システムでは非推奨ではなく明確に適用対象外とするべきである。なお、これは本節の制約事項には該当しない。本節の制約事項は、システム開発者に対する制約を示す。

がある。このようなリスクを特定し、最小化することが開発プロセスにおける優先事項でなければならない。

- 人命（ドライバーと歩行者の両方）の安全は、財産（車両と周辺環境の両方）の破壊や経済的損失よりも、あらゆる運転状況において最優先されるべきものである。しかし、不必要な衝突は避けなければならない。

7.2.8 外部品質に関する要求事項

開発する AI ソフトウェアに期待される品質要求レベルを、3 つの主要な外部品質について以下に示す。

安全性

- ASIL（Automotive Safety Integrity Level）によると、AI モジュールの好ましい安全レベルは ASIL D である [3]。ASIL D は、誤動作が発生した場合、生命を脅かす、または致命的な傷害を与える可能性を表し、これに関わる安全目標が十分で、達成されている、という最高レベルの保証を必要とする。

注：ASIL D は、深刻度、曝露確率、制御可能性の観点から、生命を脅かす（生存が不確実）または致命的な傷害を引き起こすと予期できる可能性があり、ほとんどの運転状況で物理的に発生可能であり、運転者が傷害を防ぐために何かをできる可能性はほとんどない事象と定義されている。つまり、ASIL D は、深刻度 3（S3：命の危険（生存が不確実）ないし致命的な傷害）、曝露確率 4（E4：高確率（ほとんどの運転条件で傷害が起こりうる））、制御可能性 3（C3：制御困難または制御不能）である。

パフォーマンス

- 最終的な AI システムは、ステークホルダーと開発者が合意した KPI 指標の閾値を満たす必要がある。逸脱があった場合は、報告し、十分に説明する必要がある。
- 正確さ、精度、特殊なシナリオに対する頑健性のバランスが求められている。一般的な運転状況では非常に正確でも、あまり遭遇しない、事故を起こしやすい運転シナリオに対する頑健性が低い AI モデルは推奨されない。
- あらかじめ特定した安全上重大なシナリオに対して、期待される性能レベルを満足するものであること。

公平性

- この製品・サービスの公正さについては、特定可能な要件は存在しない [4]。

---

[3] ここで検討する AI モジュール単体ではここまでの安全は担保できず、AI モジュール以外の要素で実現することになる。そのため、AI モジュールに求められる安全性のレベルは ASIL D よりもずっと低くなる。

[4] 実際には公平性は問題になる。物体検知の精度が検知対象の年齢、性別、人種、車いすへの搭乗や補助具利用の有無などに依存するべきではない。しかし、これを公平性の要求とするかについてはガイドライン第 2 版 1.5.3 節を参照。

7.2.9 外部品質特性レベルを定義する

MLQM ガイドラインにより、最終製品で確保すべき外部品質特性レベルを以下のように特定した。

表 2 実現すべき外部品質のレベル

外部品質	補足説明	想定される深刻度	実現すべきレベル
安全性 [5]	人的リスクに対する AI 安全レベル	重大な傷害。人による監視で回避可能	AISL 4
	経済的リスクに対する AI 安全レベル	かなりひどい。人による監視で回避できる。	AISL 4
AI パフォーマンス	一般的 AI パフォーマンスレベル	KPI（Key Performance Indicator）は事前に特定されるが、各 KPI の閾値は他の要因によって若干変動する可能性があり、ベストエフォートで提供される。	AIPL 2
	安全上重大な場面における AI パフォーマンスレベル	各 KPI について必要な閾値を達成し、報告値から逸脱しないこと。	AIPL 2
公平性	一般的 AI 公平性レベル	製品・サービスの公正さについて、識別可能な要件なし。	AIFL 0

7.2.10 おわりに

本節では、物体検知と場面識別を行う AI によってある程度の自動走行が可能と想定される自動車製品に関する架空のビジネス要件文書を示した。

また、要件を設定した後に、MLQM ガイドラインを一般的なビジネス要求文書にどのように取り入れることができるかを示した。これにより、AI サービス提供主体が期待する品質目標をより明確に表現しやすくなるはずである。

7.3 問題の予備的分析

本節では、AI ソリューションの機能要件と非機能要件を分析した上で導き出された重

[5] AI モジュール単体に求められる安全性レベルは低い(脚注 3 参照)ので、AISL は 1 未満とするのが妥当である。

要な仕様について説明する。

7.3.1 技術仕様

- 最終製品は、2 つの別々の機械学習モデルで構成される。
- モデルが行う学習は、教師あり学習である。
- モデルが行うタスクは、場面理解／場面識別と、画像からの物体検知である。
- 分類モデルと物体検知モデルの両方が同じデータセットを用いる。
- 2 つのモジュールはそれぞれ固有の役割を持つため、定義された品質基準は両方のタスクに関連する要件を満たしていることを確認する必要がある。
- 最終製品は、ありうるすべての気候条件、交通条件、道路条件、時間帯、照明条件の下で、自動走行中に日常的に遭遇する物体を正しく認識することが求められる。
- モデル候補としては、例えば以下のアーキテクチャが挙げられる。

タスク 1:物体検知

  - YOLOv3 [3]
  - YOLOv3 + ASFF [4]
  - YOLOv4 [5]
  - YOLOv5 [6]
  - Faster R-CNN [7]
  - M2Det [8]
  - EfficientDet-D2 [9]
  - MobileNetv1 [10]
  - MobileNetv2 [11]

タスク 2：場面識別

  - VGG19 [12]
  - ResNet50 [13]
  - InceptionV3 [14] [15]

以下、自動運転に関する入手可能な公開データセット（BDD 100k、nuScenes、KITTY など）を調査し、適性に応じて 1 つまたは複数のデータセットを選択する。

7.3.2 安全仕様

自動車安全水準(ASIL) [15]によると、ASIL D がこの最終製品に要求される安全レベルである。ASIL D は、S3、E4、C3 の組合せである。

- 深刻度分類（S）
  - S0 負傷者なし
  - S1 軽傷から中程度の傷害

- S2 重大ないし命に危険のある（生存の見込みが高い）傷害
- S3 命に危険のある（生存が不確かな）ないし致命的な傷害

- 曝露度分類（E）
  - E0 まずあり得ない
  - E1 非常に低い確率（傷害が起きうるのは稀な動作状況でのみ）
  - E2 低確率
  - E3 中程度の確率
  - E4 高確率（傷害はほとんどの動作状況で発生する可能性がある）
- 制御可能性分類（C）
  - C0 一般に制御可能
  - C1 簡単に制御可能
  - C2 普通に制御可能（ほとんどのドライバーが傷害を防ぐ行動をとれる）
  - C3 制御が難しい、または制御不可能

この BRD では、ASIL の品質基準に加えて、MLQM ガイドラインに基づく対応する品質レベルも示す。要件から判断して、AISL4 が必要な安全性レベルとなる。その他、AIPL（AI パフォーマンスレベル）、AIFL（公平性レベル）は、それぞれ AIPL2、AIFL0 となる。

7.3.3 KPI 仕様

KPI（Key Performance Indicator）は、機械学習コンポーネントからの出力が機械学習ベースのシステムを通じて達成すべき機能要件の達成度の定量的な指標である。この事例で扱う 2 つの主要な ML モデルについて、最初に考慮する KPI 仕様を以下に示す。

物体検知タスク

物体検知アルゴリズムは、通常、mAP（mean average precision）や F1 スコアなどの指標に基づいて評価される。mAP は、AI モデルの検出能力に関して、ある IoU（intersection over union）閾値に対して計算された precision-recall curve（適合率・再現率曲線）下の面積として定義されるもので、適切な候補と考えられる。言い換えれば、mAP は適合率、再現率、IoU を含んでおり、任意のモデルの検出力の指標として役立つと期待できる。

場面識別タスク

分類モデルは、通常、混同行列、正解率、適合率、再現率、特異性、F1 スコアなどの指標に基づいて評価される。**正解率**は、ある属性の属性値がバランスよく分布している場合には KPI として使える。また、属性のクラス間のバランスが悪い場合は、適合率、再現率、F1 スコアで評価するべきである。安全性が重視される自動運転車の AI アプリケーションの場合、適合率と再現率を使用することで、偽陽性と偽陰性に関するリスクの高いケースでのモデルのパフォーマンスについて、より詳細な情報を得ることができる。したがって、F1 スコアも、クラスラベルのバランスが悪い場合や高リスクのケースを扱う場合の KPI とな

り得る。

7.4 PoC（Proof of Concept）フェーズ

7.4.1 既存データセットの予備調査

上記で説明した分類タスクと物体検知タスクの AI モデルを作成するために、様々なデータセットを検討した。自動運転に関連するデータセットは多数公開されているが、その中から以下のデータセットを用いて一次検討を行った。

表 3 自動運転データセットの比較

データセット	ラベル数	画像数	天気の区別	一日の時間帯の区別	3 次元バウンディングボックス	2 次元バウンディングボックス
BDD100k	10	100000	有	有	無	有
CityScapes	30	5000	無	無	無	有
Kitti	14	14999	無	無	無	有
Semantic Kitti	28	14999	無	無	無	無
Audi A2D2 (semantic segmentation)	38	41280	無	有（タイムスタンプ）	有	有
Audi A2D2 (bounding boxes)	14	12499	無	有（タイムスタンプ）	有	有
PandaSet	28	48000	無	有（タイムスタンプ）	有	有
NuScenes	23	1400000	無	有（タイムスタンプ）	有	無
Apolloscape	25	146997	無	有（タイムスタンプ）	有	無
Canadian Adverse Driving Conditions Dataset	10	56000	無	有（タイムスタンプ）	有	無
Waymo Open Dataset	4	200000	無	無	有	有
Lyft Perception Dataset	23	450000	無	有（タイムスタンプ）	有	無
Oxford Robotcar Dataset	-	20000000	有	有	無	無

最終製品に求められる要件からみて、場面分類モデルは入力画像から天気を検出できる必要がある。BDD100k と Oxford Robotcar Dataset のみがこの要求を満たしている。しかし、Oxford Robotcar Dataset は、物体検知タスクの要件である 2D バウンディングボックスを画像に含んでいない。場面分類と物体検知の両モデルが同じデータセットを使えるようにするためには、本製品に最も適したデータセットは、BDD100k [16]と思われる。これは GitHub リポジトリ [17]から入手できる。

7.4.2 選択したデータセットの紹介

BDD100K は大規模な動画データセットとして知られている。このデータセットには、道路上の物体検知、レーンマーキング、走行可能エリアなど、様々な目的のための画像の注釈が含まれている。BDD100k は全部で 10 万枚の画像から構成されており、学習用、テスト用、バリデーション用の 3 つに分かれていて、それぞれ 70%、20%、10%の画像が割り当てられている。まずは、データセット作成者が指定したデフォルトの分割を使用する。データセットを簡単に調べると、以下のことが分かる。

(a) 概要

- 分類タスクのための属性と属性値
  - 天気：雨、雪、晴れ、曇り、一部曇り、霧、未定義
  - 場面：トンネル、住宅街、駐車場、市街路、ガソリンスタンド、高速道路、未定義
  - 時間帯：昼、夜、明け方／夕暮れ、未定義
- このデータセットには、物体を識別するためのラベルが 10 種類ある。バス、信号、標識、 人、 自転車、 トラック、 バイク、 車、 ライダー、 列車 [6]
- 以下に関する追加情報がある。隠蔽、切り取り、信号の色、車線方向、車線スタイル、車線種別 [7]

(b) 訓練用セットの情報

- 総画像数 70,000 件
- .json ファイルに情報がある画像の数 69,863 件
- 属性の数 3（天気、場面、時間帯）
- 各属性におけるカテゴリー数(未定義を除く)
  - 天気 6
  - 場面 6
  - 時間帯 3

(c) バリデーション用セットの情報

- 総画像枚数 10,000 件
- .json ファイルに情報がある画像の数 10,000 件
- 属性の数 3（天気、場面、時間帯）
- 各属性におけるカテゴリー数(未定義を除く)
  - 天気 6

[6] このリストは極めて短いため、これで訓練したモデルは例えば路上に落ちているものを検出できず、したがって安全性目標を達成する見込みがない。

[7] 同様にここには上り坂や下り坂など視界を大きく制限する場面が含まれていないので、やはり安全性目標の達成に支障が生じる。

- 場面 6
- 時間帯 3

(d) 追加情報

- 隠蔽箇所がある　真、偽
- 切り捨てがある　真、偽
- 車線方向　平行、垂直
- 車線スタイル　実線、点線
- 車線種別　横断歩道、その他の二重線、白い二重線、黄色い二重線、道路縁石、その他の一重線、白い一重線、黄色い一重線 [8]
- 信号機の色　赤、青、黄、なし
- 走行区域の種類　副候補(alternative)、主候補(direct)

7.4.3 データの分布

選択したデータセットの分布の基本的な分析結果を以下に示す。

分類タスク

BDD100k による分類タスクに関し、場面、天気、時間帯の属性値の分布は以下の通り。

表 4 BDD 100k 分類タスクの訓練用データセット統計値

Weather:

categories	clear	foggy	overcast	partly cloudy	rainy	snowy	undefined
number	37344	130	8770	4881	5070	5549	8119

Scene:

categories	city street	gas stations	highway	parking lot	residential	tunnel	undefined
number	43516	27	17379	377	8074	129	361

Time of day:

categories	dawn/dusk	daytime	night	undefined
number	5027	36728	27971	137

訓練用データセットの統計値

- 各属性の、「未定義」でない情報の比率
  - 天気　61774/69863 (88.42%)
  - 場面　69502/69863 (99.48%)

[8] このリストには車線境界だけが挙がっており、導流帯などその他の路面標識を含まないため、やはり安全性目標達成の妨げとなる。

- 時間帯　69726/69863 (99.8%)
- 少なくとも 1 つの属性が「未定義」である画像の数：8389 枚

バリデーション用データセットの統計値

- 各属性の、「未定義」でない情報の比率。
  - 天気　8843/10000 (88.43%)
  - 場面　9947/10000 (99.48%)
  - 時間帯　9965/10000 (99.8%)
- 少なくとも 1 つの属性が「未定義」である画像の数：1199 枚

表 5 BDD 100k 分類タスクのバリデーション用データセット統計値

Weather:

categories	clear	foggy	overcast	partly cloudy	rainy	snowy	undefined
number	5346	13	1239	738	738	769	1157

Scene:

categories	city street	gas stations	highway	parking lot	residential	tunnel	undefined
number	6112	7	2499	49	1253	27	53

Time of day:

categories	dawn/dusk	daytime	night	undefined
number	778	5258	3929	35

表 4 と表 5 から明らかなように、いくつかの属性値は他の属性値よりも出現頻度が高い。例えば、「時間帯」における「昼」や「天気」における「晴れ」など。また、「トンネル」のように、データセットにあまり登場しない属性値もある。このような属性値は、画像の枚数が足りず、対応が必要になる恐れがある。

物体検知タスク

このデータセットには 10 万枚の画像があり、10 のラベルが貼られている。**列車、バイク、ライダー、自転車、バス、トラック、人物、信号、標識、自動車**。表 6 と 図 18 に各ラベルがデータセット全体で何回出現しているかを示す。

表 6 訓練画像とバリデーション画像における各物体の出現回数

ラベル	訓練	バリデーション	ラベル	訓練	バリデーション
列車	136	15	トラック	29971	4245
バイク	3002	452	人	91349	13262
ライダー	4517	649	信号	186117	26885

自転車	7210	1007	標識	239686	34908
バス	11672	1597	自動車	713211	102506

明らかに、BDD100k では他のラベルに比べ、**自動車**のインスタンスが多い。このようなデータセットの均一性のなさ[9]は、検出モデルの学習時に問題となった。主に問題となったのは、検出モデルが**自動車**のインスタンスに過剰に適合してしまい、他のラベルを適切に認識できないことだった。以下の節では、この問題を解決するために考案した様々な解決策を解説する。

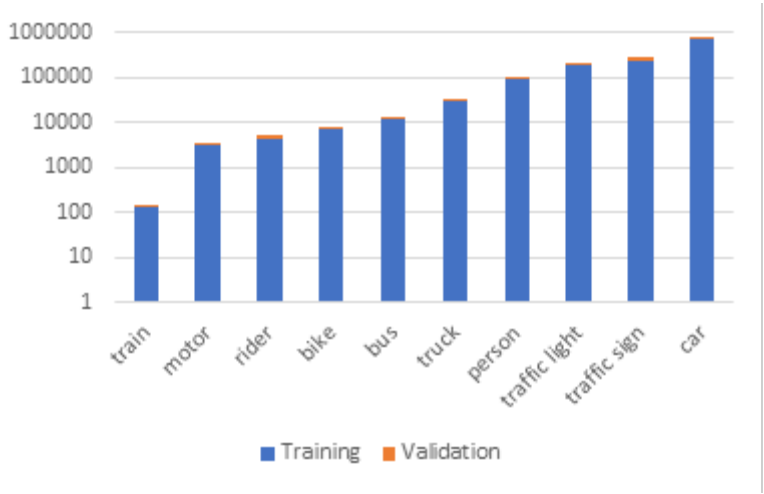


図 18 訓練画像と検証画像における各物体の出現率分布

7.4.4 候補モデルの事前訓練

PoC フェーズの目的は性能向上ではなく、利用可能なデータが候補モデルで直接使用できるかどうかを確認することである。いくつかの前処理ステップを以下に示す。

- いくつかのモデルは画像を自動的にリサイズした。それ以外の場合はリサイズしなかった。
- BDD100k のバウンディングボックス（bb）表記は、もともと、矩形の左上端の X 座標と Y 座標、右下端の X 座標と Y 座標という形式だった。この形式を COCO bb の表記形式（左上端 X 座標、 左上端 Y 座標、 幅、 高さ）に変更した。

BDD100k データセット全体に対するモデルの訓練とバリデーションを行った後、バリデーション用データセットにおける全体の mAP によって物体検知精度を測定している。分類モデルは、訓練を短時間で済ませるため、BDD100k 訓練用データセットからランダムに選択した 2 万件の画像を使用して訓練し、性能は分類精度に基づいてバリデーション用セ

[9] ガイドラインにおける「データセットの均一性」はデータセットの分布が実世界の分布と一致していることを意味するが、本章の執筆者はラベル間でのデータ量の差が小さいことを均一性と解釈している。

ットで評価した。実際の開発では、データセット全体を使用してもよい。実施者が設定した閾値以上の精度が得られた場合のみ、開発フェーズに移行し、さらなる評価・分析を行う。PoC フェーズでは、手元にあるオリジナルのデータセットと、利用可能な最新モデルの事前学習済みの重みを使用して実施した。

分類タスクのバリデーション結果

表 7 は、1 万件のバリデーション用データセットで学習した分類モデルを評価した結果である。

表 7 PoC フェーズにおけるバリデーション用セットでの分類モデルのパフォーマンス

	天気	場面	時間帯
VGG19	63.34	67.64	91.44
ResNet50	65.35	67.97	91.16
Inceptionv3	71.40	69.86	91.23

物体検知タスクのバリデーション結果

表 8 で示した結果は、1 万件のバリデーション用データセットで学習した物体検知モデルを評価した結果である。

表 8 PoC フェーズにおけるバリデーション用セットでの物体検知モデルのパフォーマンス

モデル	mAP@0.5	FPS
YOLOv3	45.7	31.25
YOLOv3 + ASFF	56.55	63.69
YOLOv4	62.3	16.07
YOLOv5	62.9	29.06
Faster R-CNN	59.3	14.58
M2Det	7.2	13.7
EfficientDet-D2	41.2	19
MobileNetv1	79.6	5.3
MobileNetv2	84.5	4.5

補足

PoC フェーズで使用したモデルはすべて実績あるアーキテクチャを持ち、オープンソース化されている。オリジナルのアーキテクチャに変更を加えていないため、各モデルの詳細（層数、ニューロン数、活性化関数など）は本稿には記載しない。以下は、物体検知モデルに使用された事前学習済みの重みである。

表 9 PoC フェーズでモデルに使用した事前学習済みの重み

モデル	重さ（URL）
YOLOv3	https://pjreddie.com/media/files/darknet53.conv.74

YOLOv3 + ASFF	ランダムな重み。
YOLOv4	https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4.conv.137
YOLOv5	https://github.com/ultralytics/yolov5/releases/download/v3.0/yolov5x.pt
Faster R-CNN	https://dl.fbaipublicfiles.com/detectron2/COCO-Detection/faster_rcnn_R_50_FPN_3x/137849458/model_final_280758.pkl
M2Det	https://drive.google.com/file/d/1NM1UDdZnwHwiNDxhcP-nndaWj24m-90L/view
EfficientDet-D2	http://download.tensorflow.org/models/object_detection/tf2/20200711/efficientdet_d2_coco17_tpu-32.tar.gz
MobileNet v1	https://1drv.ms/u/s!AvkGtmrlCEhDhy1YqWPGTMl1ybee
MobileNet v2	https://storage.googleapis.com/mobilenet_v2/checkpoints/mobilenet_v2_1.4_224.tgz

7.4.5 PoC フェーズで得られた知見

PoC フェーズに基づいて、製品の開発フェーズを開始する際の方針を決めることができる。以下、PoC フェーズで得られたいくつかの知見について説明する。

1. **明確に定義された問題領域の作成の必要性**　7.4 節で述べた利用可能なデータセットのドメイン属性には、7.2.7 節で機能要件に追加した期待されるドメイン仕様に比べて顕著な違いが見られる。例えば、物体検知モデルはあらゆる照明条件下で物体を検出することが期待されているが、今回見たデータセットには画像の明るさや照明条件を測定したような属性は存在しない。そのため明らかに既存の属性だけでは、実際のシナリオで起こりうるすべての可能な組合せを含む完全な問題領域とはみなせない。十分に定義された問題領域を作成する必要性がある。
2. **充分な被覆性と偏りのない分布の効果的なバランス**　7.4 節で見た通り、事例数の分布には偏りがある。データセットのいくつかの属性値には、対応する属性の他の属性値に比べ、非常に少ない事例しかない（例：天気の霧、場面のガソリンスタンド、物体の列車など）。そのため、問題領域を再定義する際には、属性や属性値の統合や削除の必要性を念頭に置く必要がある。また、新たに定義する領域のデータの分布が、実用・実生活における分布と乖離しないようにする必要がある。しかし、一部の問題ケースが ML モデルにとって安全性の面で非常に重要かつ重大な意味を持つことがわかった場合、その問題ケースにおけるデータの適切性を評価し、調整する必要がある。そのため、データセット分布に偏りがないことと安全上重大な事例を十分揃えることの間で、うまくバランスをとる必要がある。

3. **特殊なケースを特定する必要性**　上記の観察から、重要性の高い問題ケースを適切に特定する必要性も生じている。例えば、現実にはありえないような属性値の組合せ（例えば、夏に雪が降るなど）が存在する可能性がある。このようなシナリオは、**不可能なケース**に当たる。また、信号が青で歩行者がいるような、現実には頻繁には発生しないが、安全性の観点からは重要であり、データセットにほとんど例がないような属性値の組合せがあるかもしれない。これらは、**安全上重要な**シナリオまたは**レアケース**に相当する。このようなケースは分布分析によって明らかにして、訓練用データセットに含めるかどうかや、訓練済みモデルに期待される性能のレベルについて慎重に判断できるようにする必要がある。
4. **開発段階への指示**　7.4 節の記録からは、さらなる改良のためにどのモデルを選択するかや、KPI に設定すべき閾値を決定するなど、製品開発の次の段階への洞察が得られる。しかし、上述の特定のいくつかのケースでのモデルのパフォーマンスに関する記録は不足している。そこで、次の開発段階でモデルの性能を評価する際には、安全上重大なケースの KPI スコアも記録し、モデルが全体としても特定のケースにも必要なレベルの正しさと安定性を達成しているかどうかを確認する必要がある。

7.5 内部品質評価を伴う AI 開発

PoC フェーズでの結果に基づく予備的な分析と洞察から、今後の開発フェーズでは以下のステップを実行することが必要であることが明らかになった。

- AI 製品の要件に基づき、明確に定義された問題領域を作成すること
- 手持ちのデータを使いながら、問題領域の完全性を維持すること
- 可能な限り分布の不偏性を維持しつつ、想定されるシナリオで十分な例数を持つ訓練およびテスト用データセットを設計する
- レアケース、不可能ケースなどの一部のシナリオの特定と、その具体的な分布分析
- 学習済みモデルの総合的な性能解析と安全上重大なシナリオにおけるモデル性能の評価

本節では、MLQM ガイドラインで設定された**内部品質**特性を用いて、上記の要件をどのように満たすことができるかを示す。アジャイル開発プロセスを進める中で、これらの内部品質を確保し評価することによって、品質、安全性、性能、公平性等の要件に関して最良の AI 製品が作れるだろう。

7.5.1 A-1: 問題領域分析の十分性

問題領域分析の十分性とは、ガイドラインの 6.1.1 節にあるように、機械学習ベースの

システム が実世界で使用される場面について十分な要求分析が行われ、その分析結果が想定されるすべての場面に及んでいることを意味する。

以下では、開発時にこの品質を確保するために行った手順について説明する。ここでのポイントは以下の通りである。

1. 十分な問題領域の定義に関する詳細
2. 提案した問題領域
3. 提案した問題領域からの訓練データ例
4. 既存ドメインと提案ドメインの比較
5. 既存データセットを提案ドメインに適合させるための修正
6. 次の開発段階に向け再設計したデータセット

十分な問題領域の定義に関する詳細

まず、実生活で遭遇することが予想される自動運転シナリオの問題領域を明確に定義するために、いくつかの言葉で特徴を分析する。特徴量ツリーという概念を想定したガイドラインからヒントを得て、いくつかの**属性**とそれに対応する**属性値**で問題領域を定義する。ここでいう属性、属性値とは、ガイドラインの 2.3.6 項で定義されている意味と同じである。問題領域は、以下の点に留意して作成した。

- クライアント側の要求と既存データのラベルを徹底的に調査し、ありうるすべてのシナリオを属性とその値の組合せとして含める。
- 属性値とそのスコープを定義する際に、詳細すぎず、過小すぎず、適切なバランスを保つこと。

提案した問題領域

理想的な問題領域を作る主な目的は、自動運転車が現実に直面する可能性のあるすべてのシナリオと、ソリューション設計者に大きく関係するシナリオを網羅することである。

表 10 問題領域の仕様

Attribute	Type	Values
Perceived brightness	Numeric	0-255
Road type	Nominal	Highway, General way, Tunnel, Under FO,PL/GS, Undefined
Weather	Nominal	Fine, Cloudy, rainy, Snowy, Foggy, Heat haze, Undefined
Obstacle	Nominal	Vehicle, Others, None, Not sure, Undefined
Pedestrian	Nominal	On road, On sidewalk, None, Not sure, Undefined
Signal	Nominal	Green, Yellow, Red, None, Not sure, Undefined
Road condition	Nominal	Dry, Wet, Snowy, Undefined
Zebra crossing	Nominal	Yes, No, Not visible, Undefined
Time	Nominal	Day, Night, Dawn/Dusk, Undefined
Image clarity	Ordinal	Clear, Partly clear, Not clear, Undefined
Lighting	Ordinal	High, Normal, Low, None, Undefined
Traffic	Ordinal	High, Medium, Low, None, Undefined

表 10 は、初期の製品仕様、自動運転に関するドメイン知識、およびソリューション設計者の安全性に関する懸念を考慮しながら、このようなケースを特定する際に使用する属性と対応する値を示している。

ここで、Under FO とは、車両が高架(Flyover)や橋の下にある画像を指す。PL/GS は、駐車場やガソリンスタンドを指す。なお、属性のうち**知覚明度**は数値を取るため、その定義や計算方法について以下に補足説明を示す。

輝度／明度の算出　輝度とは、ある方向に進む光の単位面積あたりの光度を測光したものである。明度とは、客観的な輝度測定基準に対する主観的な印象を表す言葉である。比視感度(luminosity function または luminous efficiency function)は、人間の視覚的な明度知覚の平均スペクトル感度を表現する。

そこで、以下の式で画像の相対輝度を算出することが、ピクセル値から知覚的な画像の明度を算出する最も標準的な方法とされている。

$$L=0.2126R+0.7152G+0.0722B$$

- R、G、B は sRGB 値（３つのチャネルで色を表す標準的な値）から非線形関数を用いて算出される。画像全体については、全ピクセルの sRGB の平均値から R, G, B を算出する。
- 明度値の範囲は 0～255 で、0 は完全な黒、255 は完全な白である。

提案した問題領域からの学習データ例

図 19 に示した例は、データセットの各データポイントが新しい属性とその値を用いてどのように記述されるかを示している。



	0
Name	97c07e11-8f5940d7.jpg
Road type	Highway
Weather	Fine
Lighting	Low
Obstacle	None
Pedestrian	None
Signal	None
Road condition	Dry
Traffic	Low
Zebra crossing	No
Image clarity	Partly clear
Time	Night
Brightness	16.84

図 19 新しい問題領域で記述されたデータポイントの例

画像の名前は、元の BDD100k データセットに由来している。画像の縦軸と横軸は元のピクセルサイズ（1280x720）を示し、属性値は車両が遭遇した特定の自動運転状況を記述している。データセット内のすべての RGB 画像はこのピクセルサイズである。

既存データセットと提案領域の比較

表 11 に BDD100k の問題領域と提案する問題領域の比較を示す。これらの違いを観察し、データセットの再設計が必要かどうかを確認した。

表 11 BDD100k の問題領域と提案した問題領域の比較

ROAD TYPE	
Proposed	**Existing**
Highway	Highway
General way	City Street + Residential
Tunnel	Tunnel
Under FO	-
PL/GS	Gas stations + Parking Lot
Undefined	Undefined

WEATHER	
Proposed	**BDD100k**
Fine	Clear
Cloudy	Overcast + Partly cloudy
rainy	rainy
Snowy	snowy
Foggy	Foggy
Heat haze	-
Undefined	Undefined

OBSTACLE	
Proposed	**BDD100k**
Vehicle	
Others	
None	
Not sure	
Undefined	

PEDESTRIAN	
Proposed	**BDD100k**
On road	Yes
On sidewalk	Yes
None	No
Not sure	
Undefined	

ZEBRA CROSSING	
Proposed	**BDD100k (LANE TYPES)**
Yes	Crosswalk
No	Others....
Not visible	
Undefined	

TIME OF DAY	
Proposed	**BDD100k**
Day	Daytime
Night	Night
Dawn/Dusk	Dawn/dusk
Undefined	Undefined

SIGNAL	
Proposed	**BDD100k**
Green	Green
Yellow	Yellow
Red	Red
None	None
Not sure	
Undefined	

ROAD CONDITION	
Proposed	**BDD100k**
Dry	-
Wet	-
Snowy	-
Undefined	-

既存データセットを提案問題領域に適合させるための修正

本節では、既存のデータセットを提案問題領域に適合させるために行った修正を示す。

- **道路種別、天気、時間帯**：両者ともほぼ同じ用語が使われている。いくつかの属性値を統合または削除することで、BDD100k の既存のアノテーションを新しい問題領域に容易に適用できる。
  - 道路種別の**市街路**(City street)と**住宅地**(Residential)は**一般道**(General way)に統合できる。
  - **曇り**(Overcast)と**一部曇り**(Partly cloudy)は、天気の**曇り**(Cloudy)として統合できる。空の曇り度合いは明示しなくても ML プログラムが学習する。
  - BDD100k にはこの属性がないため、**陽炎**(Heat haze)は削除する。このような状況が発生する可能性は非常に低い。

- 時間帯には変更の必要はない。

- **横断歩道**：BDD100k の画像アノテーションには**横断歩道**(Crosswalk)のラベルや記述はない。しかし、BDD100k には Lane マーキングがあり、その中に横断歩道の記載があることがある。そこで、簡単な Python スクリプトを用いて、Lane マーキングの記述から必要な情報を抽出し、以下の変更を行った。
  - **横断歩道**がない場合、属性値を No とする。
  - **横断歩道**がある場合、属性値を Yes とする。
- **信号**：横断歩道属性と同様に、信号属性は、BDD100k ではアノテーション中の Traffic light ラベルに記述されている。つまり、BDD100k では信号機がある場合、信号機の色の属性が付加される。ここでも、画像中に信号機があればその属性を抽出する簡単な Python スクリプトで修正できる。画像中に複数の信号機がある場合は、信号機の属性を複数格納する。
- BDD100k には明るさの値のラベルがない。**比視感度**(人が感じる明るさ)は数値なので、データセット全体にアノテーションを施すことは容易である。そこで、Python スクリプトを用いて、データセット内の全画像の明るさを計算する。その後、テスト中の曖昧さを減らすために、明るさの値の範囲（0-255）を 5 つの等しいセクションに分割する。**非常に低い**[0-51]、**低い**[51-102]、**中程度**［102-153]、**高い**[153-204]、**非常に高い**[204-255］である。
- **歩行者**、**道路状況**、**障害物**：BDD100k では、確認できるのは画像内に歩行者がいるかどうかだけで、その位置は確認できない。しかし、歩行者の位置は非常に重要な特徴であり、無視するべきではない。車に対する信号が**青**の時、歩行者が**路上**にいるととても危険だが、**歩道**にいるのはまったく正常である。残念ながら、BDD100k データセットからこの追加情報を抽出する簡単な方法はない。同様に、BDD100k の現在の機能では、**道路状況**と**障害物**のラベルも利用できない。この問題を解決する方法としては、例えば、欠落しているアノテーションをすべて手作業で作成することが考えられる。

次の開発段階に向け再設計したデータセット

以上の議論を踏まえ、BDD100k から、訓練およびバリデーションステップで使用する AI モデルのドメインを以下のように定めた。

表 12 このアプリケーションで扱う問題領域の最終版

道路種別	時間帯	天気	歩行者	信号	横断歩道	明度
一般道	明け方・夕暮れ	晴れ	あり	青	あり	非常に高い
高速道路	日中	曇り	なし	黄	なし	高い
駐車場	夜	雨		赤		中程度
トンネル	未定義	雪		なし		低い

高架下		未定義				非常に低い
未定義						

BDD100k データセットでは、上記の属性（明るさを除く）に対するラベルが全画像にあるので、この問題領域は分類タスクの訓練とバリデーションの両方で容易に使用できる。

元の BDD100k データセットには、運転状況を表現できる 3 つの属性（**道路種別**、**時間帯**、**天気**）がある。しかし、上記の探索的なデータ分析と問題領域分析により、表 12 の通り、さらに 3 つの属性（**歩行者**、**信号**、**横断歩道**）が抽出された。つまり、PoC フェーズで示された「3 つの異なる分類タスク」の代わりに、新しい問題領域に基づいて、「6 つの異なるタスクを実行する 6 つの分類モデル」を作成することができる。運転状態の特定は、最終製品の機能要件の 1 つである。属性の追加により、運転状況をより詳しく表現でき、最終製品の最初の機能要件をよりよく満たすことができた。

明度属性は、様々な明度レベルにおいて、分類モデルと物体検知モデルの両方の性能評価で使用し、これにより、明るさの変動に対するモデルの頑健性を実証する。

歩行者の位置のラベルは、既存のデータセットでは欠落しているが、非常に重要である。また、**道路状況**や**障害物**のラベルも利用できない。しかし、（7.4 節で示したように）10 万枚の画像を手動でアノテーションするのは非常に時間がかかり、膨大な手間が必要になる。そこで、10 万枚の画像にアノテーションを施す代わりに、バリデーション用セットから「無作為抽出した 1 万枚の画像」に特製ツールを用いてアノテーションを施した。これらのアノテーションラベルは、特定の問題ケースにおいて物体検知モデルがどのように機能するかを評価するためにも使用できる。とはいえ、残りの 9 万枚の画像からの**道路状況**、**障害物**、**歩行者**の情報は、今後必要があれば作成する予定である。

7.5.2 A-2: データ設計の十分性

MLQM ガイドラインでは、問題領域分析フェーズで提示した属性値の様々なありうる組合せを確認することを推奨している。この属性値の組合せの数と詳細を検討することが、データ設計の十分性の評価の主要なテーマである。

MLQM ガイドラインでは 1.7.2 節と 6.2 節でこの品質について触れている。この段階では、システムが対応すべき様々な状況に対応した、十分な訓練データやテストデータを収集・整理するためのデータ設計の適切な検討が必要である。

以下では、開発時にこの品質を確保するための手順について説明する。ここでのポイントは以下の通り。

1. データ設計の十分性を評価する手順
2. 評価プロセスの一例
3. 特殊なケースの特定

データ設計の十分性を評価する手順

理想的には、ソリューションエンジニアは、属性値のすべての可能な組合せについて十分なデータがあることを検証する必要がある。しかし、次元数が多い問題領域では、すべての組合せを同等に考慮することはほぼ不可能である。そこで、評価の容易さと品質管理の望ましいレベルとのバランスを保つために、以下の手順が考えられる。

- すべての属性とそれに対応する値について、可能な組合せの総数を見積もる。適切な数の属性を用いて、属性値のありうる組合せを見積もる。
  - 数が多くない場合は、それぞれの組合せに対して十分なデータがあるかどうかを確認する必要がある。
  - 数が非常に多い場合は、最も重要で、データセットの大部分をカバーする組合せを優先して評価する必要がある。
- 属性とその値の組合せでケースを定義し、最も重要なケースと無視できるケースを区別する。

評価プロセスの一例

以下に、評価プロセスの例を示す。簡単のため、BDD100k トレーニングセットから抽出した、同様の分布を持つが、より小さい約 2 千件の注釈付きデータからなるデータセットを使用する。もし、データセット内の全ての画像に対して適切なラベルを見つけることができれば、この評価はデータセット全体に対して行われるべきである。

考慮した問題領域の詳細

ここでは、状況の複雑さを評価するため、以下の属性とそれに対応する値を考える。

全属性数 7 個　全属性値数 34 個

- 明度：高い、低い、正常（3 個）
- 道路種別：一般道、高速道路、駐車場又はガソリンスタンド、トンネル、未定義（5 個）
- 道路状態：乾いている、降雪あり、濡れている、未定義（4 個）
- 信号：青、赤、黄、なし、不明、未定義（6 個）
- 歩行者：路上、歩道上、なし、不明、未定義（5 個）
- 障害物：車、その他、なし、不明、未定義（5 個）
- 天気：晴れ、曇り、雨、雪、霧、未定義（6 個）

評価結果の概要

以下は、より小規模だが同様の分布を持つ 2 千件のデータセットで定量分析を行った結果である。

- 1 属性だけの組合せの数＝全属性値の数＝34 個
- 7 属性と全属性値からなる組合せの数= 54,000 個

組合せの総数が膨大になるので、7 つの属性から組合せ一つあたり 2 つの属性を取ること（ペアワイズ分析）を考えてみよう。

- 組合せを確認する属性ペアの総数：21 個
- 属性値の組合せの総数：542 個

21 通りの組合せから、**明度＋信号**を例にとってみる。このペアには、18 通りの属性値の組合せが存在する。表 13 は、その 18 件のリストと、各件のデータの有無を示している。

表 13 照明＋信号の組合せのデータ有無

明度	信号	個数	比率
高い	青	25	1.23
高い	なし	70	3.44
高い	不明	7	0.34
高い	赤	4	0.20
低い	青	345	16.94
低い	なし	375	18.42
低い	不明	41	02.01
低い	赤	77	03.78
低い	未定義	1	00.05
低い	黄	11	00.54
通常	青	326	16.01
通常	なし	626	30.75
通常	不明	41	02.01
通常	赤	78	03.83
通常	未定義	1	00.05
通常	黄	8	00.39
高い	黄	0	0
高い	未定義	0	0

また、上記の組合せにおけるデータ分布を以下のグラフに示した。

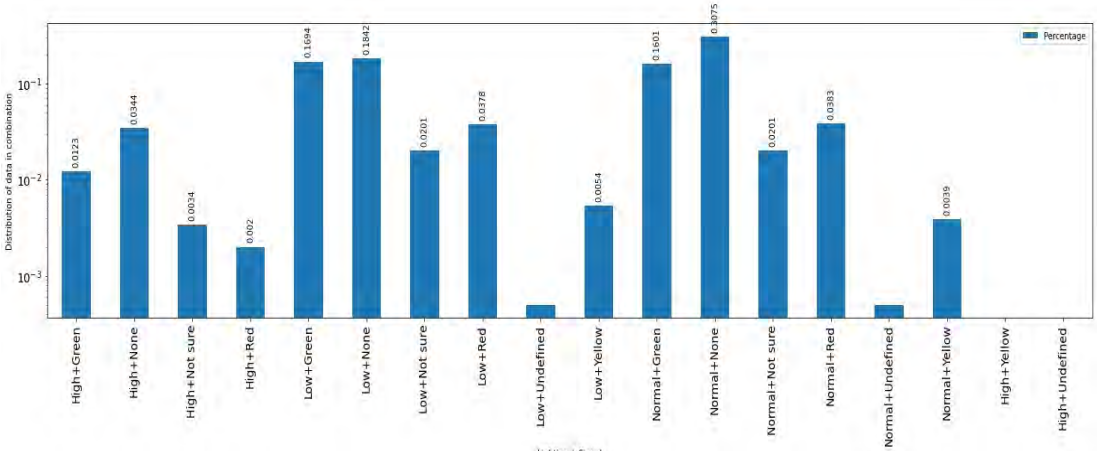


図 20 照明＋信号の組合せ で起こりうる事例のデータ分布

このように被覆性を確認する作業は、明らかに非常に手間暇がかかる。そこで、システムが特に警戒するべき特別なケースのみを特定し、それらのケースについて被覆性を評価することにした。

特殊なケースの特定

ここでは、重要度の高い特定のケースをいくつか設定し、そのケースのみについてデータの有無を調査することで、およその完全性を確認することができる。これらの重要なケースを注意深く選ぶことで、データセットの大部分をカバーしていることを確認することができる。

重要なのは、特別な注意が必要なケースである。例えば、現実にはありえないが、データセットには何らかの形で存在する属性値の組合せがありうる。このような例は除外することにしてもよい。また、AI モデルのパフォーマンスが低下するようなケースもありうる。事故を防ぐには、そのような危険なケースについて十分なデータがあるかどうか、しっかり確認する必要がある。

ここでは、問題領域における不健全なケース（実世界ではありえないケース）とリスクの高いケース（モデルの性能劣化の可能性があり、より高い安全性を保つべきケース）を明示的に特定することを目標とする。

不健全なケース

不健全なケースは、ソリューション設計者が事前に特定し、訓練から除外する必要がある。このようなケースを説明するために、ある属性（**一次条件属性**と呼ぶ）にいくつかの値を設定し、他の属性の値と組合せて、作成した組合せが現実に存在し得るかどうかをチェックする。そうして見つけた組合せを表 14 に示す。これらの組合せは、現実には（この特定のアプリケーションと問題領域では）起きないはずなので、これらの組合せに当たるデータの有無をチェックし、もしあれば除外する必要がある。

表 14 データには含まれないはずの不健全なケース

ケース	一次条件属性	一次条件値	二次条件属性	二次条件値
0	天気	雪	道路状況	乾いている
1	天気	雨	道路状況	乾いている
2	道路種別	高速道路	信号	青
3	道路種別	高速道路	信号	赤
4	道路種別	高速道路	信号	黄
5	道路種別	高速道路	横断歩道	はい
6	道路種別	高速道路	歩行者	路上
7	道路種別	高速道路	歩行者	歩道上

安全上重大な／高リスクケース

以下に特定した危険なケースを示す。これらのケースを危険とみなす理由は、AI モデルのパフォーマンスが低下する可能性のある場面に関わっていて、それらの場面で AI モデルが誤った決定を下すと危険な状況に陥る可能性があるからである。

2 つの属性を 1 つのグループとする組合せ

1. 道路種別：高速道路＋天気：雨
2. 道路種別：高速道路＋時間：夜
3. 天気：雨＋時間：夜
4. 道路種別：一般道＋天気：雨
5. 道路種別：一般道＋歩行者：路上
6. 道路種別：一般道＋時間帯：夜
7. 天気：雨天＋歩行者：路上
8. 天気：雨＋時間帯：夜
9. 歩行者：路上＋時間帯：夜

3 つの属性を 1 つのグループとする組合せ

1. 道路種別：一般道＋天気：雨＋歩行者：路上
2. 道路種別：一般道＋天気：雨＋時間：夜
3. 道路種別：一般道＋歩行者：路上＋時間帯：夜
4. 天気：雨＋歩行者：路上＋時間帯：夜

分布を計算した後、被覆性の基準または**閾値**を設定する必要がある。閾値は、具体例をよく観察してうまく設定すれば、より重要性の高い組合せ（すなわち、危険なケース）に存在するデータの数が十分であるかどうか推定する際の基準値となり得る。次の開発段階では、閾値と実際の値との比較に基づいて、以下の問題について決定する必要がある。

- データ拡張などを用いてより多くのデータを用意するべき組合せがあるか

- 完全に無視すべき組合せがあるか
- データがない組合せにどう対処するべきか

7.5.3 B-1: データセットの被覆性

MLQM ガイドラインでは、前項で設計した全ての**対処が必要な状況の組合せ**に対して、十分なデータ（特にテストデータ）が与えられていることを**データセットの被覆性**と定義している。

この特性軸を設定する目的は、要求分析やデータ設計で特定された状況やケースにおいて、データ不足による学習不足やデータの偏りによる特定条件での学習漏れが発生しないことを保証することにある。

以下では、開発時にこの品質を確保するための手順について説明する。ここでのポイントは以下の通り。

1. データセットの被覆性を評価する手順
2. 評価プロセスの一例
3. 記録した結果からの洞察

評価に必要な手順

各重要場面、特に 7.5.2 節で示したケースにおいて、十分な量のデータが利用可能かどうかは、以下のアプローチで確認できる。

- 各グループに存在するデータポイントの数を確認する。
- 各グループで
  - 閾値以上のデータポイントを持つ組合せ数を見積もる
  - データポイントが閾値より大幅に少ない組合せの数を見積もる
  - データのない組合せの数を評価する

以上の数値から、必要に応じて、以前にデータの被覆性のために定義した閾値の見直しを行うことができる。必要であれば一部の製品機能を断念することも考える。

評価プロセス例

前述したように、AI 製品の本来の開発の次のステップでは、ペアワイズ分析ではなく、7 つの属性とすべての値（54000 件）の可能な組合せの分布を計算することが考えられる。また以下の分析は、バリデーション用データセットに対しても行う必要がある。しかし、ここで扱っているのは架空の製品なので、評価プロセスの説明を簡単にするため、7.5.2 節で使用したのと同じデータセットについて、特定のケースについてのみ、データセットの被覆性の評価を続けよう。以下は、前節で述べた様々なケースについて被覆性を調査した結果である。

不健全ケースにおけるデータの有無

表 15 は、不健全な場合のデータの存在を示している。

表 15 不健全なケースでのデータ被覆性

ケース	一次条件属性	一次条件値	二次条件属性	二次条件値	データセット内比率(%)
0	天気	雪	道路状況	乾いている	0
1	天気	雨	道路状況	乾いている	0.098
2	道路種別	高速道路	信号	青	1.031
3	道路種別	高速道路	信号	赤	0.295
4	道路種別	高速道路	信号	黄	0.049
5	道路種別	高速道路	横断歩道	はい	0.344
6	道路種別	高速道路	歩行者	路上	0.098
7	道路種別	高速道路	歩行者	歩道上	0

前述したように、学習データはこのような不健全なケースを含むべきでない。このようなデータが含まれるのには、アノテーションの誤り、ラベルの付け間違いなど、さまざまな理由が考えられる。このようなケースはデータセットから除外してもよい。もしこのようなケースでデータ数が非常に少ない場合は、(合理的に最小限の労力で) 追加調査を行ってエラーの原因を調べ、可能であればラベルを修正することができる。

危険なケースでのデータの存在

7.5.2 節で述べたリスクの高いケースについて、データの被覆性の調査結果を以下にまとめる。

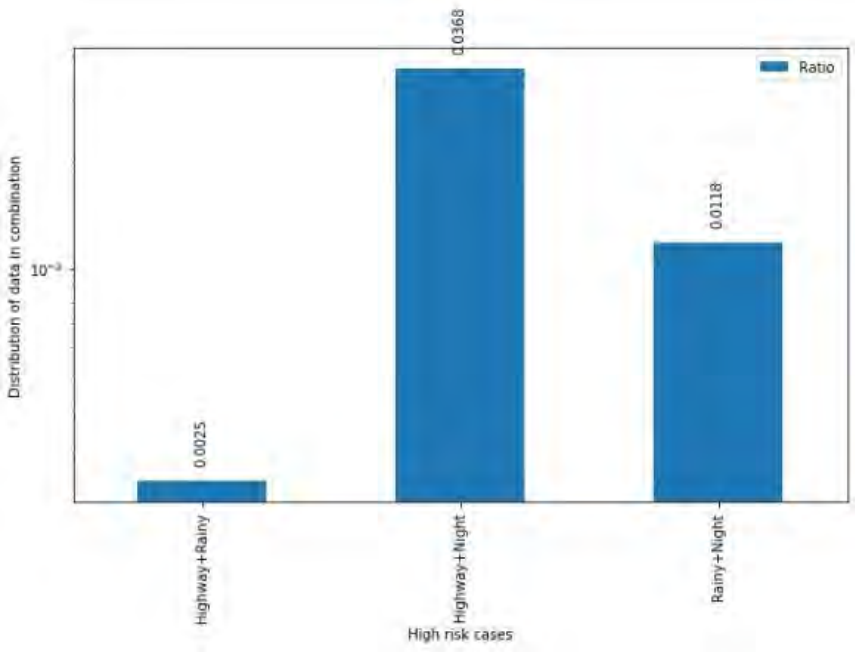


図 21 いくつかのリスクの高い事例に対する被覆性

グループ 1：2 つの属性を取る組合せ

1. **道路種別：高速道路＋天気：雨**
2. **道路種別：高速道路＋時間：夜**
3. **天気：雨＋時間：夜**

グループ２：２つの属性を取る組合せ（続き）

1. 道路種別：一般道＋天気：雨
2. 道路種別：一般道＋歩行者：路上
3. 道路種別：一般道＋時間：夜
4. 天気：雨＋歩行者：路上
5. 天気：雨＋時間: 夜
6. 歩行者：路上＋時間帯：夜

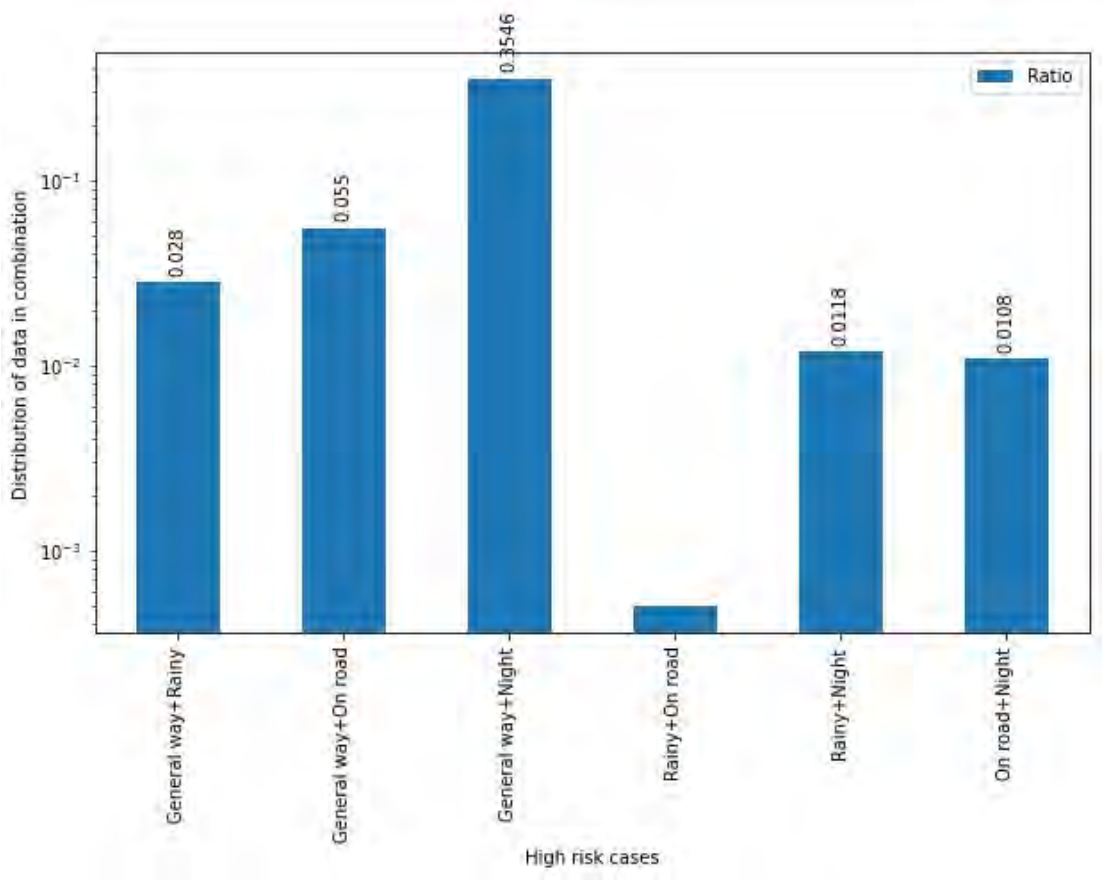


図 22 いくつかのリスクの高い事例におけるデータ被覆性（続き）

グループ３：３つの属性を取る組合せ

1. 道路種別：一般道＋天気：雨＋歩行者：路上
2. 道路種別：一般道＋天気：雨＋時間：夜
3. 道路種別：一般道＋歩行者：路上＋時間帯：夜
4. 天気：雨＋歩行者：路上＋時間帯：夜

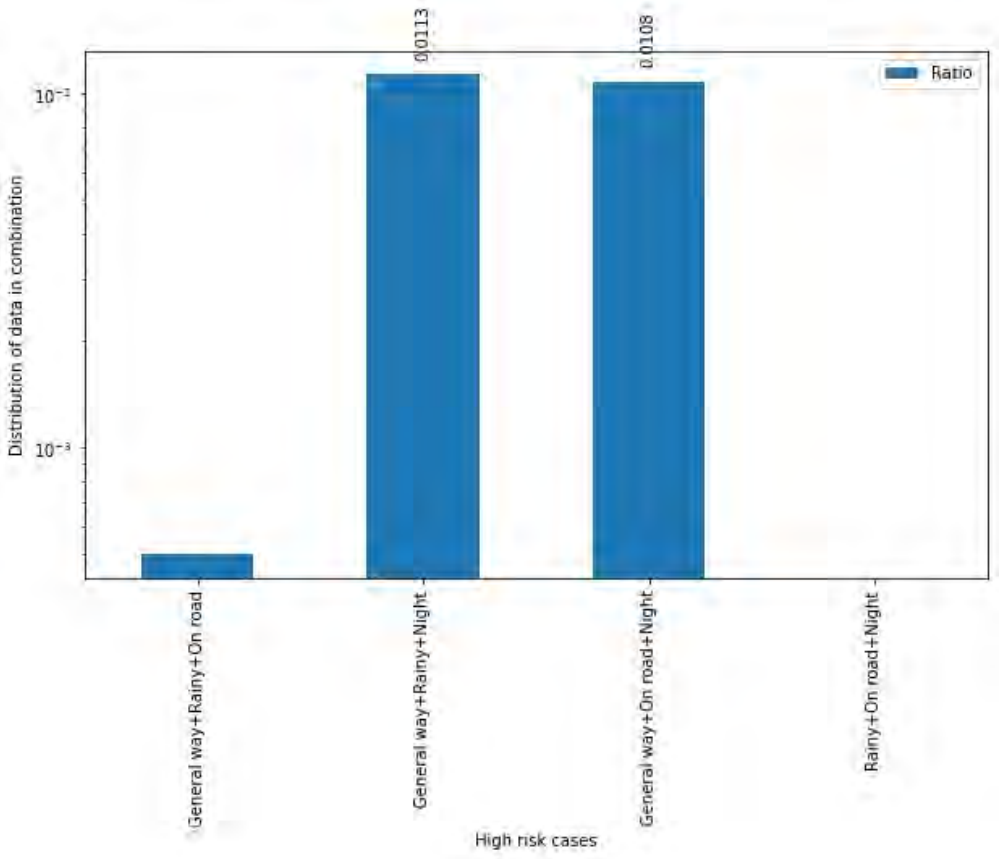


図 23 いくつかのリスクの高い事例におけるデータ被覆性（続き 2）

記録した結果からの洞察

被覆性分析の結果から、ある組合せには他の組合せよりも著しく少ない例しかないことが確認された。このような分布は実際の状況でも予想されることだが、これらの特定の場面では、データ不足のためにモデルの性能が低下する可能性がある。このようなケースに対して閾値を設定することで、稀なケースの被覆性を高めるために、追加データが必要なケースを選別することができる。本検討では、閾値を 100 画像に設定した。

このような場合にもモデルの性能が低下しないように、データセットに稀な入力を適量用意する必要がある。前節では、データセットにあまり出現しないが、自動運転シナリオにとって極めて重要な希少画像を特定した。この種の希少な画像は十分な数を学習に取り入れる必要があるため、これらの入力をたっぷり用意する方法を検討する必要がある。一つの方法は、このようなコーナーケースを生成するためにデータ拡張プロセスを使用することである。実際、適切な拡張技術を適用すれば、敵対的データを生成して学習済みモデルの頑健性と安定性をチェックすることも可能である。

BDD100k データセットでは、BDD 動画データセットから、目的のクリップからより多くのフレームをサンプリングすることで、類似したシナリオの例をより多く抽出する方法

もある。BDD100k データセットの画像は全て BDD 動画より取得したものである。さらに、BDD100k の各画像は、BDD 動画データセットでどの動画が使用されたかを指し示す ID を持っている。そのため、BDD100k の画像の動画が見つかれば、その画像に関連する画像をさらに見つけることができる。BDD 動画データセットの全ての動画フレームはラベル付けされており、python スクリプトで抽出することができる。

例えば、ノイズに強い学習済みモデルを作るために、フロントガラスに付着した水による不鮮明な画像を含めたい場合、同じ動画からタイムスタンプの異なる画像をいくつか使用することができる。データの被覆性調査から、そのようなケースは稀にしか起きないことが分かっている。そこで、以下の対応が可能である。まず、動画データセットからフロントガラスに付着した水による不鮮明な画像を特定する。次に、特定した画像より 1 秒前のフレームと 1 秒後のフレームの 2 枚を抽出する。これら 3 つの画像は、道路位置が異なるだけで同じ外乱を持つことになる。動画データセットには、BDD100k と同じアノテーションを用いて全ての物体がラベル付けされている。そのため、抽出された画像は訓練やバリデーションのためのデータセットに含めることができる。こうして、利用可能な画像が少ない組合せの画像枚数を増やすことができる。



(a) 1 秒前　　(b) 元画像　　(c) 1 秒後

図 24 BDD 動画 02701fba-809c39f3.mov の連続フレーム

7.5.4 B-2: データセットの均一性

MLQM ガイドラインでは、**データセットの均一性**は、前節で述べた**被覆性**と対になる概念とされている。データセットの**均一性**を評価するためには、データセットの元の分布を調べ、データに偏りがないかを検査する必要がある。データセット中の各ケース内の分布が、入力データ全体における出現頻度に従っている場合、そのデータセットは**均一**であるとみなされる。ここでは、**被覆性**と**均一性**のバランスを評価することが第一の関心事である。MLQM ガイドラインでは、被覆性と全体の均一性のどちらを優先させるか、そのバランスをどうとるかを考える必要があるとしている。

以下では、開発時にこの品質を確保するための手順について説明する。ここでのポイントは以下の通り。

1. データセットの均一性を評価する手順

2. 評価プロセスの一例

評価に必要な手順

上記の優先順位は、モデルの性能に大きく影響する。与えられた要件に応じて、ソリューション設計者は以下を選択することができる。

- モデルの総合的な性能を優先させる。この場合、データセット中にごく稀にしか発生しないようなケースがあった場合、十分なデータがないため、ML モデルはそのケースを適切に学習できない可能性がある。
- あるいは、たとえデータが十分でなくても、より重要性の高い特定のケースに対するモデルの性能を優先させる。この目的を達成するためには、特定のケースを取り上げ、そのケースの自然発生頻度を無視して、このケースのデータを多く入れようとすることになる。その結果、特定の性能は向上しても、全体の性能は悪化する可能性がある。

上記のシナリオを考慮して、以下を行う必要がある。

- 本来の発生頻度、すなわちデータセットの分布を評価し、特定のケースの被覆性を向上させるために講じた措置が、データセットに偏りをもたらしていないかどうかをチェックする。
- 有意な偏りが見つかった場合、データのより少ない部分を考慮するか、他の類似の出所からのデータを考慮して、データセットの期待分布を計算することができる。そして、期待される分布と偏った分布とを比較する。
- 問題ケースの分布のずれをどの程度許容するかは、先に述べた要件と優先度に基づいて設定できる。

評価プロセス例

均一性分析の評価手順を示すために、データの簡単な部分を考えてみよう。実際の製品開発では、別の出所からの別のデータセットと比較できるケースについて出現頻度を測定できるよう、訓練用とバリデーション用のデータセット全体の分布を測定する必要がある。例えば、本検討では、問題領域は約 2 千個のデータを含むデータセットと以下の属性で表される。

- **障害物**：なし、車両、その他、わからない、未定義
- **歩行者**：なし、わからない、歩道上、路上、未定義
- **道路種別**：一般道、高速道路、橋の下、ガソリンスタンド、未定義、トンネル
- **明度**：高い、正常、低い

データセットの全般的な分布の評価

まず、様々な属性ごとに、その属性値に渡る、全般的データ分布を評価しよう。次のグラフは、**明度**属性のデータが属性値間でどのように分布しているかを示している。

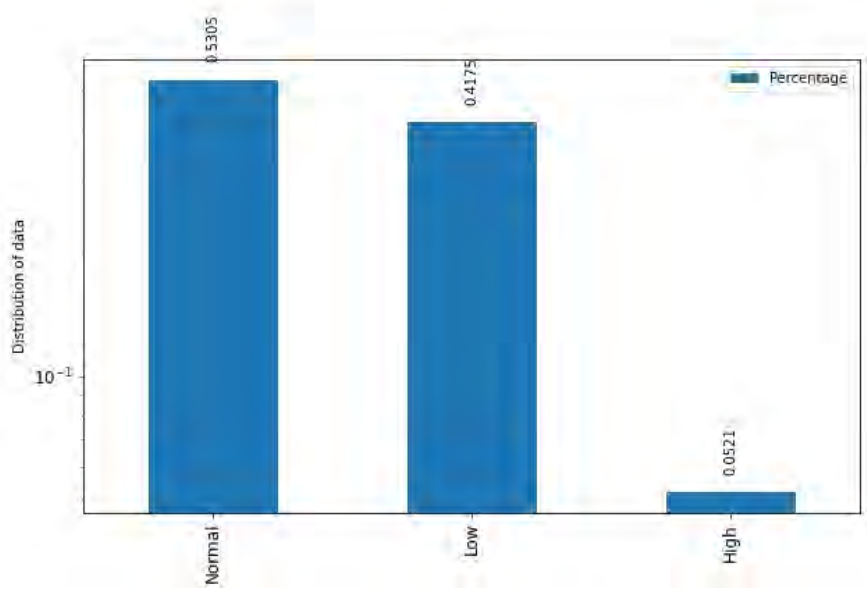


図 25 属性明度の属性値間のデータ分布

以下のグラフは、属性**道路種別**のデータについて、属性値間の分布を示している。

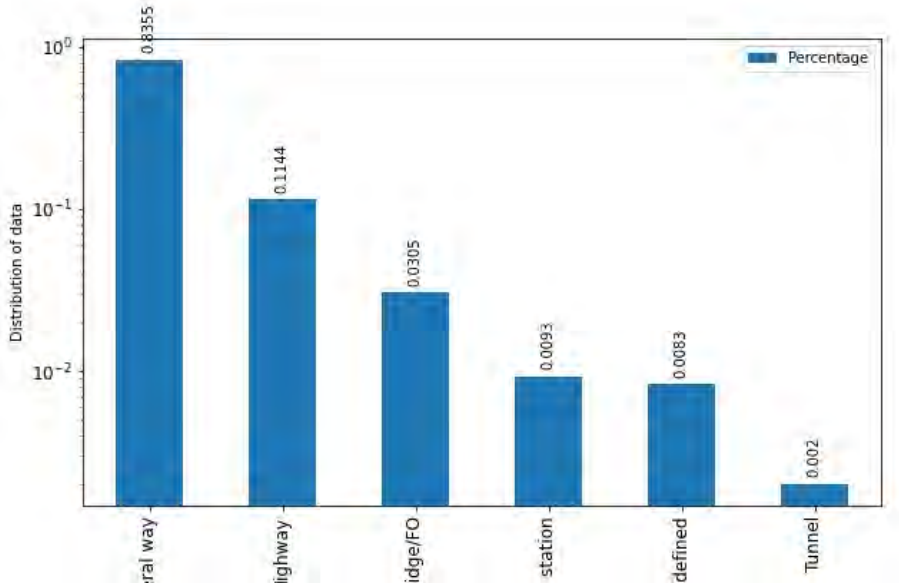


図 26 属性道路種別の属性値間のデータ分布

別の出所から、同様の分布を持つ 6 千個のデータからなる別のデータセットが入手できると仮定しよう。この第 2 のデータセットの分布は、比較のための参考または想定分布として考慮することができる。

表 16 **明度**の属性値における出現頻度の比較

属性値（明度）	元のデータセットの分布	第 2 データセットの分布	比率の差
通常	53.05	62.15	-9.1
低い	41.75	35.07	6.68
高い	05.21	02.75	2.46

表 17 **道路種別**の属性値の出現頻度の比較

属性値	元データセットの分布	第 2 データセットの分布	比率の差
一般道	83.55	81.84	01.71
高速道路	11.44	13.45	02.01
橋又は高架下	03.05	02.9	00.15
駐車場又はガソリンスタンド	00.93	00.84	00.09
未定義	00.83	00.72	00.11
トンネル	00.2	00.26	-00.06

観察された差異に基づいて、許容できるずれの閾値を設定できる。その後、データ拡張その他のデータ分布の操作により、何らかの属性値に偏りが見つかった場合は、その偏りが許容できるずれの範囲内かどうかをチェックできる。

組合せケース間のデータ分布の評価

同様の比較は、様々な属性の組合せ、特にリスクの高いケースなど特別な意味を持つケースを含めて行う必要がある。リスクのあるケースはデータセットの中で稀かもしれず、その場合偏ったデータセットを敢えて作成するかどうかの判断が必要になる。そこで、均一性と被覆性のバランスを取るために、重要度の高い特定の組合せについて、上記のような詳細な調査を行う必要がある。**照明＋道路**タイプの組合せ属性に関する同様の分析を以下に示す。

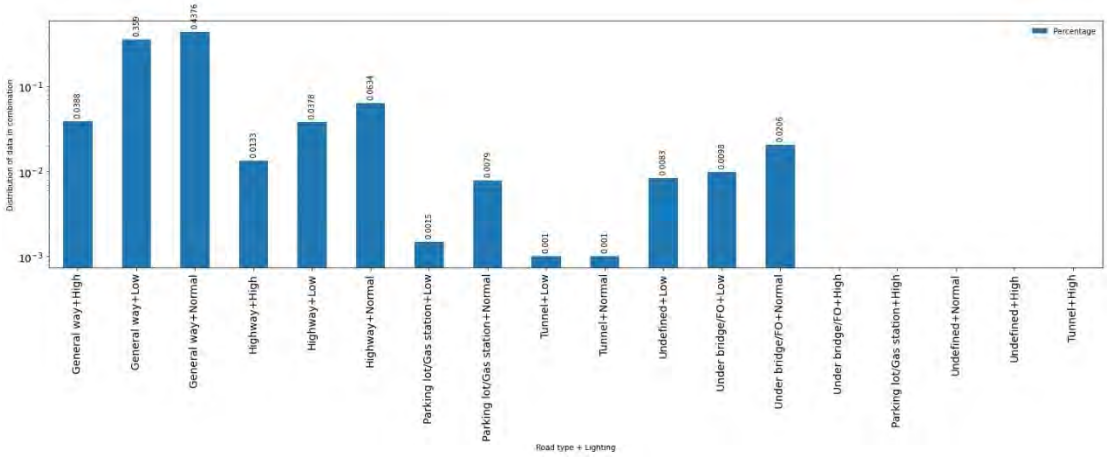


図 27 **道路種別＋照明**の属性値の組合せのデータ分布

同様の比較は、想定分布と実際の分布の違いについても行うことができる。

表 18 属性値の組合せの出現頻度の比較

道路種別	明度	元のデータセット分布	第 2 データセットの分布	比率の差
一般道	高い	3.88	2.05	1.83
一般道	低い	35.9	29.08	6.82
一般道	通常	43.76	50.72	6.96
高速道路	高い	1.33	0.7	0.63
高速道路	低い	3.78	4.45	0.67
高速道路	通常	6.34	8.28	1.94
駐車場又はガソリンスタンド	低い	0.15	0.12	0.03
駐車場又はガソリンスタンド	通常	0.79	0.72	0.07
トンネル	低い	0.1	0.09	0.01
トンネル	通常	0.1	0.17	-0.07
未定義	低い	0.83	0.58	0.25
橋又は高架下	低い	0.98	0.75	0.23
橋又は高架下	通常	2.06	2.15	0.09
橋又は高架下	高い	0	0	0
駐車場又はガソリンスタンド	高い	0	0	0
未定義	通常	0	0.12	0.12
未定義	高い	0	0	0
トンネル	高い	0	0	0

ただし、実際の運用環境からの実データが入手でき、十分な時間があるなら、これらの評価手順は、最終的な AI 製品の訓練とバリデーションに使用する実世界のデータセットで実行するべきである。これは、リファレンスガイドの次版における将来作業として考えることができる。この版のリファレンスガイドでは、ほとんどの場合、詳細な実験や正確な評価手順を示していない。むしろ、評価プロセスやモデル開発プロセスがどのように行われるかを論じている。また、物体検知タスクか分類タスクの一方の分析のみを行っている場合もある。放置したタスクについても同様の結果を得ることが望ましい。今後、アジャイル型開発プロセスの各ステップにおいて、PoC 結果、評価結果、モデル性能のより体系的な記録方法を使うことが望まれる。

7.5.5 B-3: データの妥当性

この版のリファレンスガイドでは、内部品質 B-3：**データの妥当性**の評価手順を示していない。次版の課題である。

7.5.6 C-1: 機械学習モデルの正確性

機械学習モデルの正確性とは、学習データセット（訓練用データ、バリデーション用データ、テスト用データからなる）の入力に対して、機械学習コンポーネントが意図したとおりに機能することを表す言葉である。MLQM ガイドラインでは、学習の収束性や訓練データの質（例えば、データセットに外れ値や誤ったラベルが少ない）もこの概念に含む。

ここでのポイントは以下の通り。

1. PoC フェーズに基づく決定事項
2. 物体検知モデルの正しさの評価手順

PoC フェーズに基づく決定事項

7.4 節の PoC フェーズで示した結果をもとに、50%以上の mAP を達成し、リアルタイム画像で動作可能なモデル（FPS が 15 以上のもの）のみを実験対象として継続することにした。その理由は、50%mAP 未満のモデルは、50%mAP 以上のモデルに比べ改良に多くの時間を費やす恐れがあるためである。FPS については、モデルの FPS を向上させることは可能であるが、FPS を向上させると精度が低下する可能性がある。そのため、今回の評価例では、精度の向上のみに注力し、FPS の向上や検討は今後の課題として残すことにした。Yolov3, M2Det, EfficientDet-D2, MobileNetv1, MobileNetv2 は、最小 mAP を達成できなかったり FPS が低すぎたりするため、実験から除外した。残りのモデル(Yolov3+ASFF, Yolov4, Yolov5, Faster R-CNN)をこの参考例の対象とした。これらの精度を改善して、ASIL D の実現に十分な mAP を達成することを目指す。

物体検知モデルの正しさの評価手順

データセットに含まれるノイズ情報を除去する

図 28 の画像は、いくつかのバウンディングボックスが重なり合い、学習プロセスを困難にしている様子を示している。この問題は、各検知モデルの学習過程にノイズをもたらし、得られる精度を低下させる可能性がある。

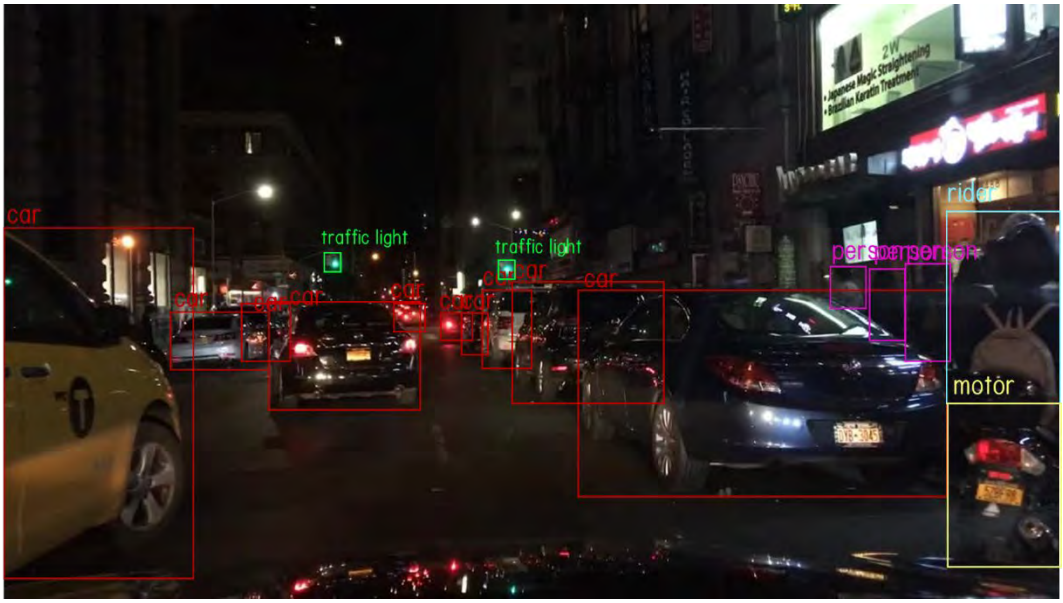


図 28 バウンディングボックスの重ね合わせのある画像例

精度を上げるための適切な解決策は、ボックスの数を減らし、車に近い物体を学習させることかもしれない。しかし、車両からの物体の距離を計算することは、別のレベルの複雑さを追加し、深度推定が必要になるかもしれない。バウンディングボックスの数を減らすために、**ボックス削減**（と呼んでおく）アプローチを取ることにしよう。

ここでやることは、バウンディングボックスが恣意的に設定したウィンドウより小さいか等しい場合、そのバウンディングボックスをトレーニングデータセットから削除する。そして、それらのバウンディングボックスを除いた新しいデータセットを作成し、検出モデルの再トレーニングに使用する。ウィンドウの設定は以下の 6 種類である。

0：すべての物体を使用する

10: 10x10 ピクセル内の物体は予測やバリデーションに使用しない。

20: 20x20 ピクセル内の物体は予測やバリデーションに使用しない。

30: 30x30 ピクセル内の物体は予測やバリデーションに使用しない。

40: 40x40 ピクセル内の物体は予測やバリデーションに使用しない。

50: 50x50 ピクセル内の物体は予測やバリデーションに使用しない。



図 29 余計なバウンディングボックスの除去に使うウィンドウのサイズ

表 19 ボックス削減を適用後の mAP の比較

モデル	データセット全体	10x10 を削除	20x20 を削除	30x30 を削除	40x40 を削除	50x50 を削除
YOLOv3+ ASFF	56.55	58.43	63.13	65.11	53.00	40.74
YOLOv4	62.3	64.17	69.70	72.93	58.78	45.15
YOLOv5	62.9	64.57	70.62	73.68	59.87	45.43
Faster R-CNN	59.3	60.66	66.14	70.41	54.58	41.08

表 19 は、バウンディングボックスを削除し、選択した各検出モデルを再トレーニングした後の物体検知モデルの性能を示している。10x10、20x20、30x30 以下のバウンディングボックスを削除した場合、それぞれ約 1.7%、7.1%、10.2%の mAP の増加が見られる。しかし、40x40、50x50 のバウンディングボックスを削除すると、それぞれ 3.7%、17.1%の減少が見られる。この結果から、検出モデルの学習過程にノイズをもたらすバウンディングボックスが存在することが示された。このメカニズムは、他のデータセットにも適用して、重複するアノテーションを削減し、mAP が増加するかどうかを確認することができる。

表 20 ボックス削減を適用後のラベル mAP 結果

Yolov4 mAP(%)	データセット全体	10x10 を削除	20x20 を削除	30x30 を削除	40x40 を削除	50x50 を削除
信号	51.92	52.65	59.86	66.84	47.25	31.73
標識	66.47	67.65	74.15	76.64	60.78	46.46

車	79.25	78.26	84.16	86.9	71.62	59.84
人	51.15	55.84	60.18	62.52	49.71	38.26
バス	63.13	64.58	68.92	71.87	60.86	43.27
トラック	47.81	50.63	57.27	60.24	45.32	32.61
ライダー	61.78	62.42	66.58	69.36	58.75	46.97
自転車	72.43	75.68	79.91	81.34	69.64	57.31
バイク	67.5	69.84	76.62	80.85	64.93	50.18
全体の mAP	**62.3**	**64.17**	**69.70**	**72.93**	**58.78**	**45.15**

表 20 は、Yolov4 で**ボックス削減**アルゴリズムを使用した場合のラベルごとの mAP 結果である。この結果は、バウンディングボックスの一部を削除することが、検出モデルがすべてのラベルの精度を向上させるのに役立つことを示している。

検知モデルの特定訓練

また、特定の属性値の画像を用いて検知モデルを訓練することで、精度を向上させることができる。これにより、特定の状況に対してうまく機能するように検知モデルを訓練することができる。BDD100k では、**道路種別、天気、時間帯、歩行者、信号、横断歩道、明度**の 7 種類の属性が設定されている。しかし、本実験では、ASIL D の要件に照らして優先度の高い**道路種別、天気、時間帯**の属性のみを使用した。

検知モデルの学習には、少なくとも 2 万枚の画像が必要であると想定する。図 30 は属性値ごとの画像枚数を示しており、実験が可能な属性値は、**市街路（道路種別）、昼（時間帯）、夜（時間帯）、晴れ（天気）**だけなのがわかる。

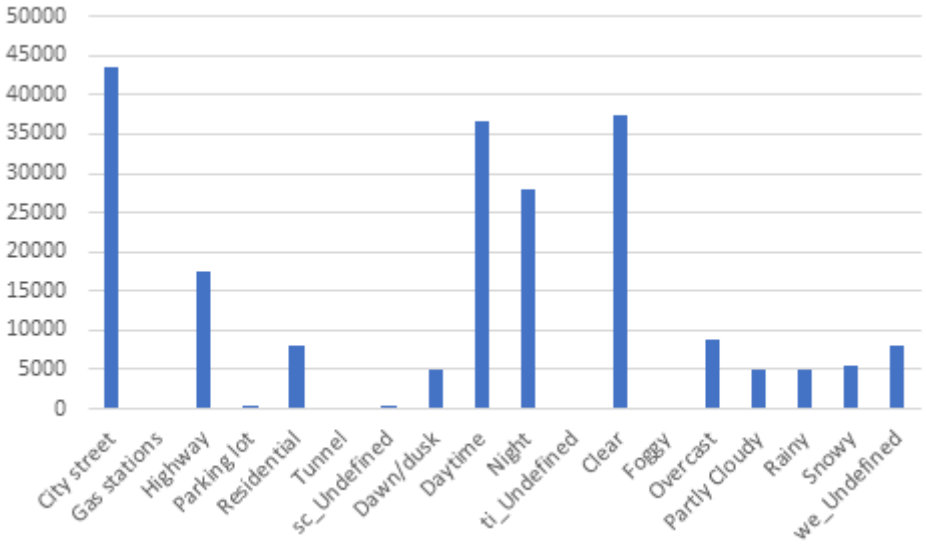


図 30 属性値ごとの画像数

図 31 は、各検知モデルを**市街路**、**昼**、**夜**の属性値を持つ画像のみで学習させた場合と、データセット全体を用いて学習させた同じ検知モデルとの mAP 比較である。バリデーション用データセットの画像を用いた mAP の計算では、特定の属性値の画像のみを用いている。その結果、すべてのケースで改善が見られる。改善幅は、**昼**について平均 10%と高く、**夜**は 6%と最も低い。

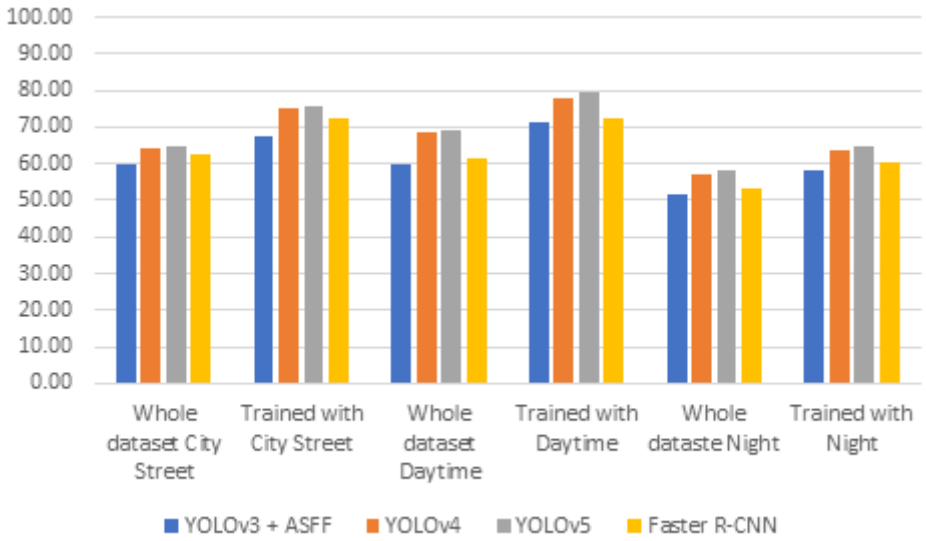


図 31 属性値を用いた特定訓練と
特定訓練に用いた属性値のデータセット全体との mAP 比較

まとめると、この実験で、多様な状況のデータセットを使うのではなく、特定のデータセットを使って学習した方が良い状況があることが実証された。これを考慮すると、例えば、車が市街路にある場合は**市街路**の属性値についてうまく動くモデルや、それぞれ**昼**と**夜**にうまく動く 2 つのモデルを用意するなど、複数の検知モデルを使い、状況に応じて使い分けるのがよいかもしれない。唯一の必要条件は、車が市街路にあるのか、今が昼なのかを検知する仕組みが必要になることである。今回の実験では、**ボックス削減**アルゴリズムと特定状況の訓練は組合せておらず、以降の版のリファレンスガイドの課題である。また、将来的には、**昼**の**市街路**でうまく動作するモデルなど、属性値の組合せも検証する予定である。

7.5.7 C-2: 機械学習モデルの安定性

MLQM ガイドラインによれば、**機械学習モデルの安定性**とは、学習データセットに含まれない、あるいは学習データセットに十分に類似しない入力データに対して、機械学習要素が適切な反応を示すことを意味する。この品質を確保するためには、モデルの汎化能力、コーナーケース/レアケースへの対応、敵対的データに対する性能などを評価する。これらの問題に対するモデルの頑健性やモデルの安定性を評価することは、アジャイル AI 開発サイクル中の単一手順からなるプロセスではない。むしろ、AI 開発のライフサイクルのさま

ざまな段階で、さまざまな手法や手段を取り入れるのがよい。

ここでの議論のポイントは以下の通り。

1. 評価に必要な手順
2. 汎化能力の評価
3. 敵対的な画像に対する頑健性の評価

評価に必要な手順

この内部特性を評価するために、実行可能な手順を以下に説明する。

- **汎化能力を評価する。**実世界では、入力データには多くのバリエーションが存在する。そのすべてを訓練に含めることは不可能である。そのため、問題領域全体の入力データを学習させ、訓練用データセットの近傍ではモデルが適切に動作することを期待することが重要である。次に、モデルが過学習する傾向を抑制する必要がある。最後に、モデルの汎化能力は、訓練期間とバリデーション期間中にモデルが遭遇したことのない入力に対してテストしなければならない。また、使用した評価手法の有効性についても分析する必要がある。
- **ノイズや敵対的データに対する頑健性を評価する。**入力データには常にノイズが付加される可能性がある。そのため、ノイズ処理能力や、懸念される特定のノイズに対する頑健性も測定する必要がある場合がある。頑健性の評価方法と頑健性のレベルは、敵対的データに対するモデルの応答を反映するように記述される必要がある。

汎化能力の評価

このセクションでは、nuImage [18]という別のデータセットを使用して、BDD100k を用いて訓練した検知モデルの性能を評価する。nuImage の概要は以下の通り。

表 21 nuImage データセットの概要

ラベル数	画像数	天気がある	時間帯がある	3 次元バウンディングボックス	2 次元バウンディングボックス
23	93476	いいえ	はい（タイムスタンプ）	はい	いいえ

このデータセットには 93476 枚の画像が含まれており、訓練用データ、テスト用データ、バリデーション用データに分けられている。

表 22 nuImage データセットの画像分布

	画像数	データセットに占める割合(%)
訓練用	67279	71.97462
バリデーション用	16445	17.59275
テスト用	9752	10.43262

ただし、ラベルが全くない画像もある。

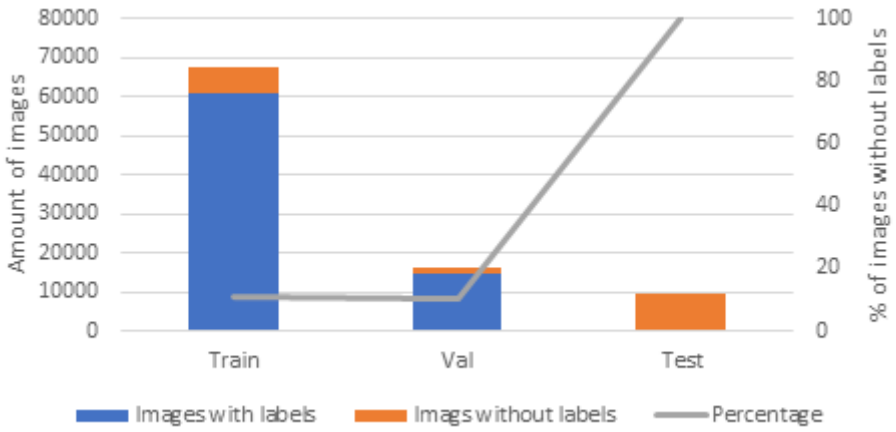


図 32 nuImage でのアノテーションがある画像とない画像の分布

そのため、検知モデルの学習に使える画像は 60,668 枚、バリデーションに使える画像は 14,884 枚にとどまる。

注釈の数については、BDD100k はラベル 10 個のデータセットであるのに対し、nuImage にはラベルが 23 個ある。検知モデルの汎化能力を評価するために、BDD100k と nuImage に共通するラベルの正解情報を用いている。そこで、23 個のラベルを BDD100k で学習した検知モデルに合うものに絞り込んだ。**表 23** ではその結果を緑で示した。BDD100k のラベルのうち nuImage でも検知できるのは、**自転車、バス、車、バイク、人、トラック**だけである。

表 23 nuImage ラベルの分布と BDD100k ラベルへの変換に使用したラベル(緑)

ラベル	訓練	バリデーション	合計	比率
動物	173	82	255	0.04
人.歩行者.大人	121200	28721	149921	21.61
人.歩行者.子供	1683	251	1934	0.28
人.歩行者.工事作業者	10465	3117	13582	1.96
人.歩行者.モビリティ	1828	453	2281	0.33
人.歩行者.警官	368	96	464	0.07
人.歩行者.ベビーカー	293	70	363	0.05
人.歩行者.車椅子	33	2	35	0.01
可動物.障壁	70112	18433	88545	12.76
可動物.瓦礫	2461	710	3171	0.46
可動物.カート	3030	645	3675	0.53
可動物.パイロン	69016	18587	87603	12.63
固定物.自転車ラック	2461	603	3064	0.44
車両.自転車	13708	3352	17060	2.46

車両. バス. 多両編成	203	62	265	0. 04
車両. バス. 一両編成	6538	1823	8361	1. 21
車両. 乗用車	202809	47279	250088	36. 05
車両. 工事車両	4768	1303	6071	0. 88
車両. 緊急. 救急車	34	8	42	0. 01
車両. 緊急. 警察	104	35	139	0. 02
車両. バイク	13682	3097	16779	2. 42
車両. トレーラー	3285	486	3771	0. 54
車両. トラック	29456	6858	36314	5. 23

結果を比較するために、2 種類の Yolov4 のモデルを訓練した。1 つは BDD100k トレーニングデータセットを用いて訓練したモデル、もう 1 つは BDD100k アノテーションにラベルを変換した nuImage トレーニングデータセットを用いて訓練したモデルである。BDD100k で訓練したモデルは、データセット外の画像に対して検知モデルがどの程度効果的に動作するかを評価するために使用し、もう一方のモデルは、同じデータセットで学習した場合の違いを知るための正解として使用する。図 33 はこの比較を示したものである。**自動車、バス、トラック**のラベルは同程度の精度を達成し、その差は 5%未満である。**バイク**は、モデル間の差が最も大きい（約 16%）。これは、BDD100k には**ライダー**というラベルもあり、モデルが**バイク**ではなく**ライダー**だけを検知することがあるためである。

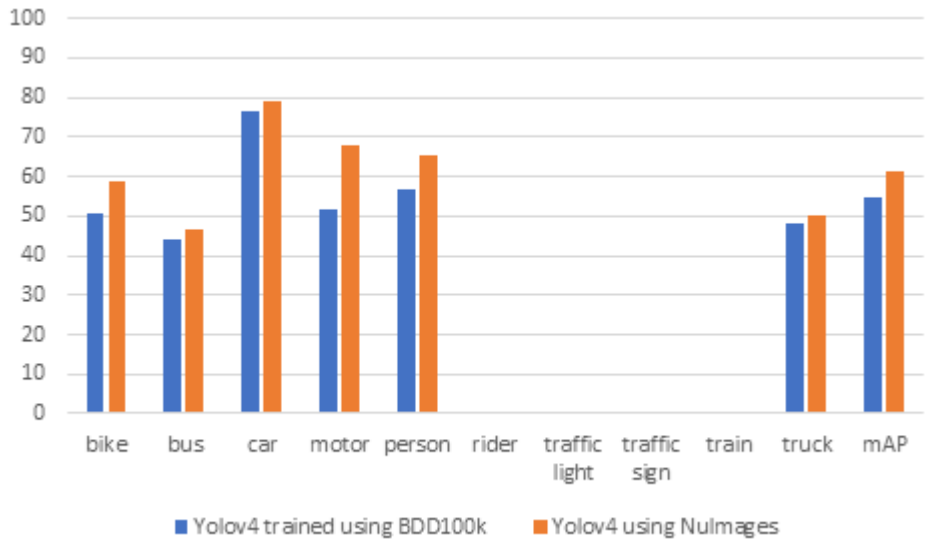


図 33 BDD100k と nuImage の訓練用データセットで訓練した Yolov4 の nuImage バリデーション用データセットに対する精度の比較

両モデルの mAP を比較すると、BDD100k は 54.64%、もう一つのモデルは 61.32%を達成した。この差は 7%未満である。したがって、Yolov4 は BDD100k で学習した後、データセット外の画像に対しても良好な汎化能力を達成している。

敵対的な画像に対する頑健性の評価

物体検知モデルの頑健性を評価するために、Surprise Adequacy [19](付録 B を参照)を使用する。敵対的データを生成するには、特定のノイズパターンを導入するか、敵対的攻撃に利用可能な技術を用いる必要がある。最近の文献によると、敵対的攻撃の代表的な技術として、以下のようなものがある。

- Fast Gradient Sign Method (FGSM)
- Basic Iterative Method（BIM-a、BIM-b）
- Jacobian-based Saliency Map Attack (JSMA)
- Optimization-based attack（Opt）

この例では、FGSM を使用し、他の 3 つの敵対的な攻撃は今後の課題とした。

物体検知モデルに敵対的攻撃を行う上での困難

懸念事項の一つは、ニューラルネットワーク(NN)アーキテクチャに対する敵対的攻撃の実施に関する最新の文献（FGSM、BIM-a、BIM-b、JSMA、Opt（C&W））でも、従来の畳み込みニューラルネットワークしか扱っていないことである。これらの敵対的攻撃実施例では、NN の最終層を使用して敵対的データを生成する。標準的な最先端の物体検知モデルは、そのアーキテクチャがユニークで複雑である。特に、最後の数層の構造が多様であるため、これらの利用可能な方法を用いて敵対的データを生成することは非常に困難である。また、Yolo の出力予測は、他の NN とは異なる。他の NN では、通常、出力層は各ラベルのスコアを与えるが、Yolo は全てのバージョンで、3 つの検知テンソルからなる 3 並列出力層を持っており、それぞれに独自の前ボックスがあり、それぞれが前の 2 倍の解像度を持つのである。非最大抑制法を用いた後、ラベルスコアの高いボックスだけが残される。敵対的データの作成には、非最大抑制法は、選択法であって層ではないので使用できない。このため、Yolo ニューラルネットワークを用いた敵対的データの生成は、本検討で解決する必要があった課題の一つとなった。

試してみたアプローチ

FGSM 攻撃を YOLOv4 アーキテクチャに適用しようとしたが、FGSM 攻撃は複数の並

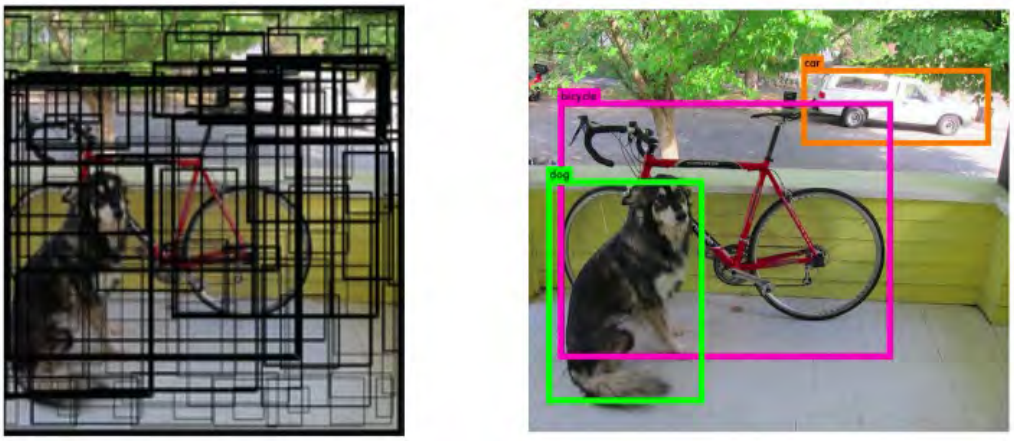


図 34 Yolo 最終層の出力例

列出力層を持つ NN では機能しないため、失敗した。この問題を解決するために、以下のさまざまなアプローチを試した。

1. Tensorflow-Keras から Pytorch に**フレームワークを変更した**。
2. YOLOv3、YOLOv5 などの異なるアーキテクチャを使用した。
3. 見つかったラベルの信頼度だけを維持して他のラベルを 0 にするように**予測出力を修正した**。問題は、他のラベルが全て信頼度 0 であり、敵対的データを決定するために閉じていないため、どのラベルの信頼度が最も近いかを判断できず、敵対的データを生成できないことである。
4. TensorFlow の tf.gradients を使わず**手動でグラデーションを生成した**。
5. BIM-a や JSMA など、さまざまな敵対的攻撃を用いる。

見込みのある解決策

これまで述べた解決策はすべてうまくいかなかった。しかし、敵対的データを作成したり、コーナーケースを検知したりするのに役立つ、他の解決策を考案した。

- Inception や MobileNet など、**別の NN モデルを使用する**。これらのモデルには、Yolo のような最終並列層の問題はない。試しに、MobileNetv2 を使って FGSM の敵対例を生成してみたが、完璧に動作した。これらのモデルに Surprise Adequacy を適用することにも、特に問題はない。したがって、別のモデルを用いてコーナーケースを決定し、その後、そのコーナーケースを Yolo で使用することが、適切な解決策になり得る。

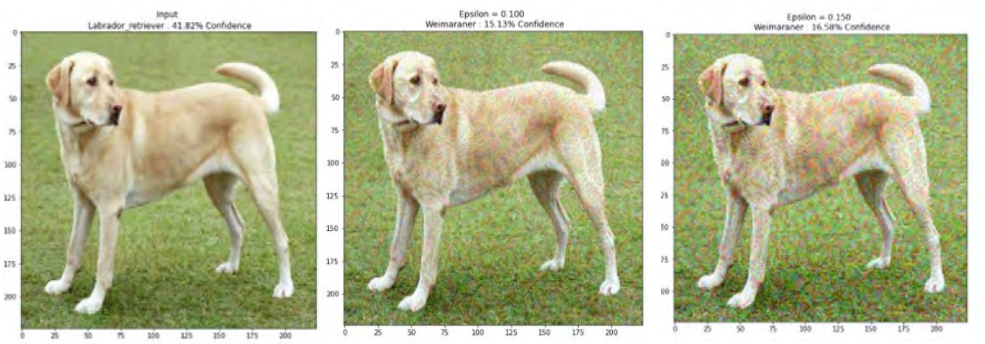


図 35 FGSM による MobileNet への攻撃の実装に成功した例

- **さまざまな攻撃を試してみることが、**課題解決につながることもある。しかし、Surprise Adequacy を行うために敵対的データを生成する方法はすべて出力ラベル信頼度を用いるため、NN が複数の最終出力層を持たず、1 つの最終出力層を持つ場合にのみ有効である。そのため、これまで述べたものとは異なる新たな攻撃を定義することが必要かもしれない。例えば、後述する **1 ピクセル変更法**を用いて、誤ったラベリングを引き起こす割合が高い**コーナーケース**を検知することなどが考えられる。

異なる NN を用いた敵対的攻撃の適用

訓練済み検知モデルの頑健性を評価するため、MobileNetv2 を用いて敵対的データを作成し、Surprise Adequacy を用いて頑健性を検証することにした。ここでは、MobileNetv2 は BDD100k と Tensorflow-Keras フレームワークを用いて訓練した。学習後、学習済みモデルは 84.49%の精度を得ることができたが、FPS は 4.5 だった。FPS が低いため、MobileNetv2 を自動運転車の検知モデルとして使用することは考えられなかった。しかし、MobileNetv2 を用いて FGSM のような敵対的な攻撃を生成することは可能である。敵対的データを生成した後、Surprise Adequacy を用いて、コーナーケースを検証する。これらのコーナーケースは、すべての検知モデルで同じであると仮定されており、すべての検知モデルで同じように扱う。以下の画像は、MobileNetv2 と FGSM を用いて生成した敵対的データの例である。

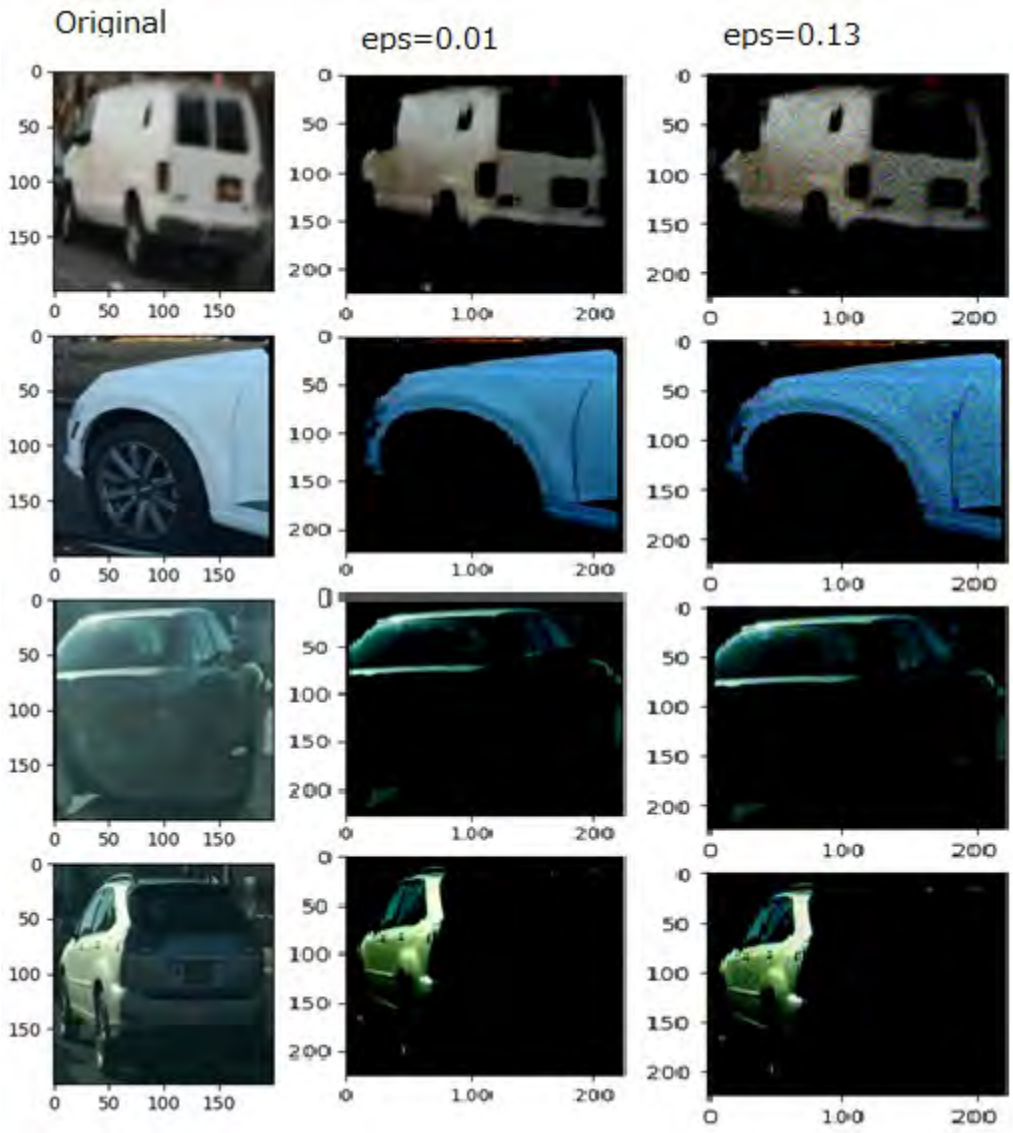


図 36 BDD 100k データセットで FGSM 攻撃を使用して生成したデータ
(eps = 0.01 & 0.13)

MobileNetv2 への Surprise adequacy の適用

本節では、Surprise Adequacy [20]を用いて、訓練データのコーナーケースを検知し、訓練用データセットからそれらを削除し、以下のすべての検知モデルを再訓練する。Yolov3 + ASFF, Yolov4, Yolov5, Fast R-CNN, MobileNetv2.

BDD100k では、訓練用データセットに 1,286,871 個のバウンディングボックスが存在する（データセットには 69,863 枚の画像がある）。**表 24** は、 Surprise 攻撃に使用したバリデーション用データにおけるバウンディングボックスの分布である。

表 24 BDD 100k のバウンディングボックス・ラベル分布

ラベル	BB の数
自転車	7210

バス	11672
車	713211
バイク	3002
人	91349
ライダー	4517
信号	186117
交通標識	239686
列車	136
トラック	29971

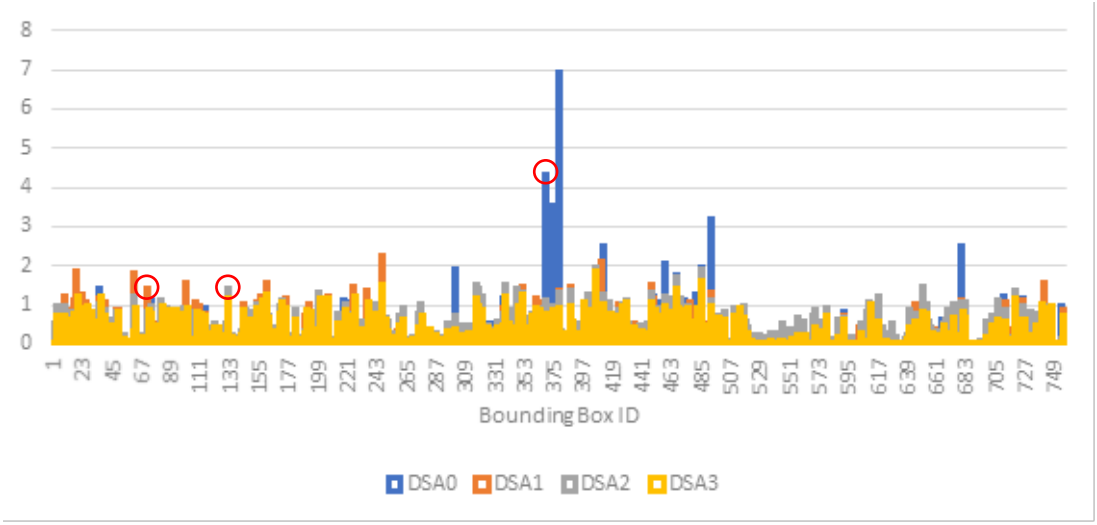


図 37 訓練データ中の自転車の最初の 750 個の
バウンディングボックスの Surprise Adequcy

自動車のバウンディングボックスは多すぎる。しかし、**列車**を含めすべてのラベルのコーナーケースを検出するのに十分な数のバウンディングボックスが存在する。これは、100 枚の画像があれば、Surprise adequacy がラベル間のコーナーケースを判断するのに十分だからである。

図 37 は、訓練データ中の**自転車**の最初の 750 個のバウンディングボックスの surprise adequacy DSA_0, DSA_1, DSA_2, DSA_3 を計算した結果を示す。計算した DSA のいずれかで距離が 1.5 より大きい場合、そのバウンディングボックスはコーナーケースであると定義する。例として、3 つのコーナーケースを図 37 では赤丸でマークし、図 38 にも示した。



ID	DSA0	DSA1	DSA2	DSA3
73	1.54	0.82	0.97	0.79



ID	DSA0	DSA1	DSA2	DSA3
131	0.52	0.39	1.6	1.15



ID	DSA0	DSA1	DSA2	DSA3
368	4.32	0.98	1.11	0.78

図 38 Surprise adequacy で検出された自転車のコーナー事例

BDD100k の全ラベルをチェックした結果を以下に示す。

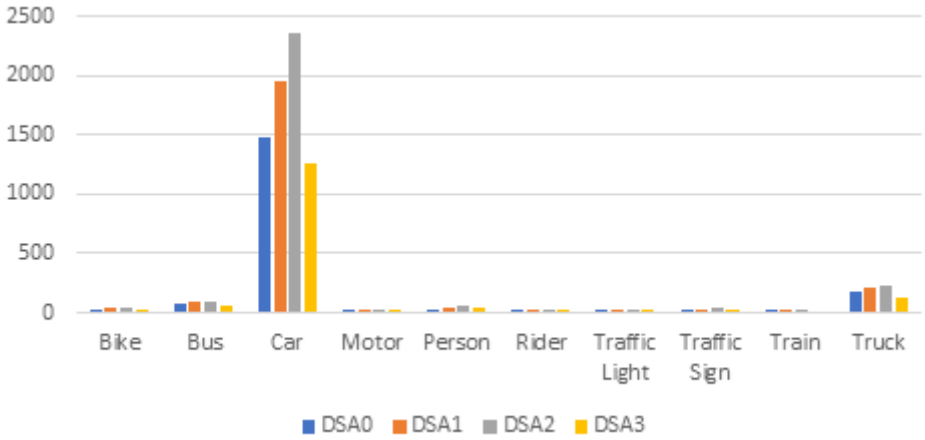


図 39 ラベルごとに検出されたコーナー事例の件数

上のグラフの主な問題点は、BDD100K には他のラベルに比べ、**自動車**のラベルが多すぎることである。その結果、各ラベルについて DSA の影響を判断することが困難になっている。

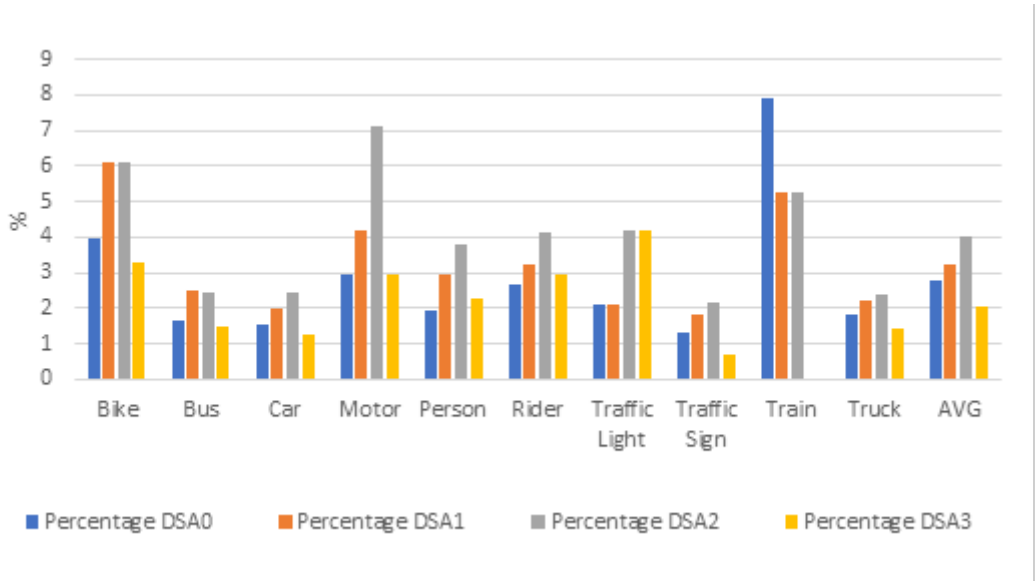


図 40 BDD 100k データセットにおけるコーナー事例の割合

しかし、ラベルごとに何個のコーナーケースが検出されたかをパーセンテージで表すと、ラベルによって特定の DSA が他の DSA よりも有効なことがわかる。例えば、DSA_2 は、ラベル**バイク**(Motor)に対してより多くのコーナーケースを検出することができる。全体として、DSA_3 は、BDD100K のコーナーケースをあまり検出できていない。例えば、ラベル**列車**のコーナーケースを全く検出できていないのである。

検出したコーナーケースを利用して、6 種類の MobileNetv2 を訓練した。

- 各 DSA ごとにモデル 1 つを、元のデータセットからその DSA で検出したすべてのコーナーケースを除いたもので訓練した。
- 元のデータセットから、4 つの DSA のいずれかによって検出されたすべてのコーナーケースを除いたものでモデル 1 つを訓練した。
- 元のデータセット全体でモデル 1 つを訓練した。

ここで使用したモデルは、ImageNet データセットから画像を識別するために訓練済みの MobileNetv2 である。これにより、訓練が高速になり、最終層のみを再訓練するだけで済む。

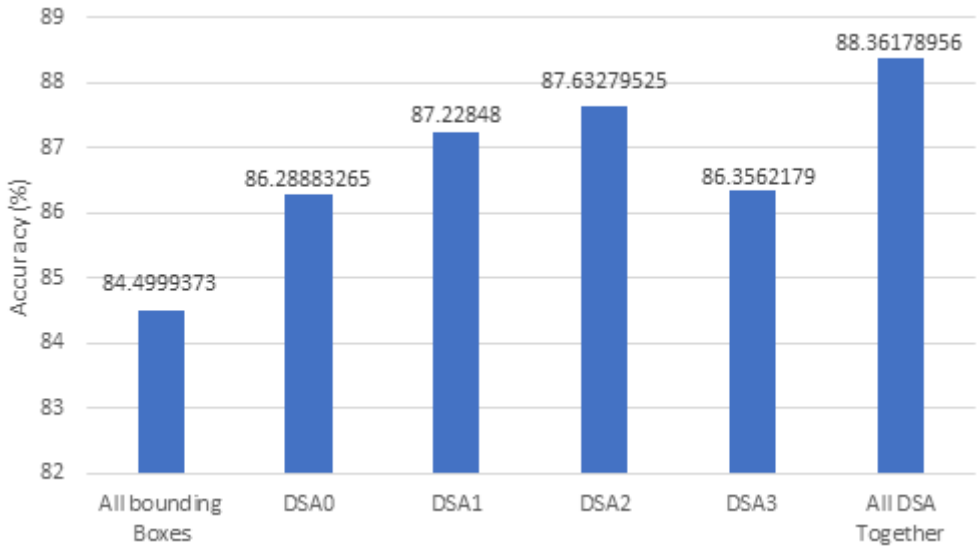


図 41 MobileNetv2 によるコーナー事例除去後のモデル精度

画像を削除せずに訓練した場合（All bounding Boxes）、MobileNetv2 は 84.5%の精度を得ることができた。すべての DSA で精度は向上したが、向上の程度には違いが見られた。DSA_2 では、全画像を使用した場合よりも 3.13%向上し、最も高い結果(87.635)が得られた。これは、DSA_2 が最も多くのコーナーケースを検出しているためと思われる。ここで、任意の DSA がコーナーケースとして検出した画像をすべて削除し、モデルを再訓練するとどうなるだろうか。結果は、All DSA Together である。この場合、全画像を学習させたモデルに比べ、3.86%の改善が見られる。これらの結果から、Surprise adequacy はコーナーケースを検出することができ、それらを訓練データセットから削除すれば、精度が向上することが分かる。

次の作業は、学習データセットから検出されたコーナーケースを削除し、Yolov4 を再訓練することである。これを行うのは、MobileNetv2 の FPS が 15 未満であり、安全上の理由から自動運転検知（ASIL D）に使用できないからである。このため、Yolov4 を用いて、Surprise adequacy と MobileNetv2 によって検出されたコーナーケースが他の検出方法に使えるかどうかを確認する。

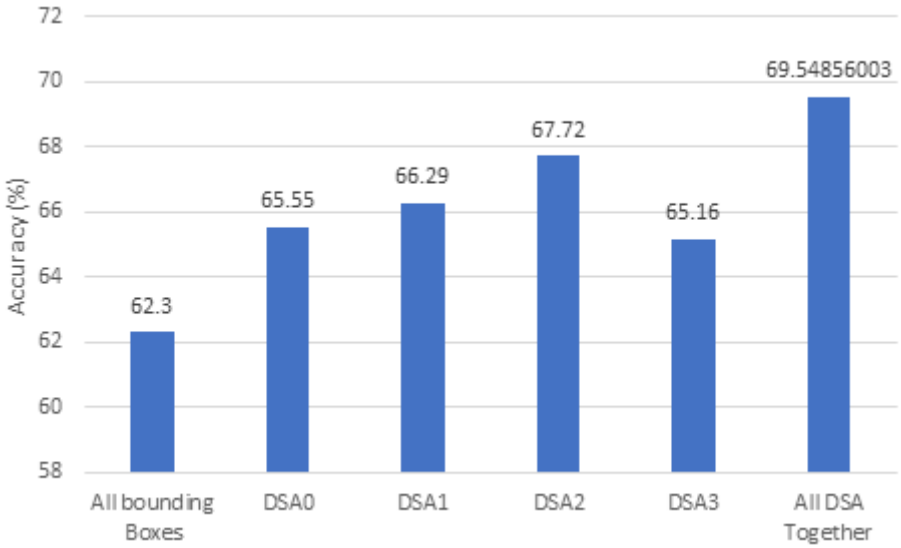


図 42 Yolov4 によるコーナー事例除去後のモデル精度

図 42 は、MobileNetv2 で検出したコーナーケースを訓練用データセットから削除し、Yolov4 を再訓練した結果である。コーナーケースを除去して学習させた全てのモデルで、精度の向上が見られる。この増分は、MobileNetv2 の結果で得られたものと同様である。これは、Surprise adequacy が 1 つのニューラルネットワークを使用してコーナーケースを検出することができ、そのコーナーケースは別のニューラルネットワークの頑健性を高めるために使用できることを実証している。

なお、今回は精度を上げるために Surprise adequacy を使用しているが、Surprise adequacy の本来の役割はモデルの頑健性を高めることである。本節の例で精度向上が見られたのは、BDD100k にある**自動車**のラベルが多すぎ、検出されるコーナーケースの数が最も多いのが**自動車**であるためである。一般には、コーナーケースがデータセットから削除され、訓練プロセスから情報が失われる結果、頑健性が低下する可能性がある。しかし、BDD100k では、コーナーケースを除去することで、検出モデルがより確実に**自動車**を検出できるようになった。このような結果になったのは、**自動車**とそれ以外のラベルの境界が取り除かれているからである。

1 ピクセル変更

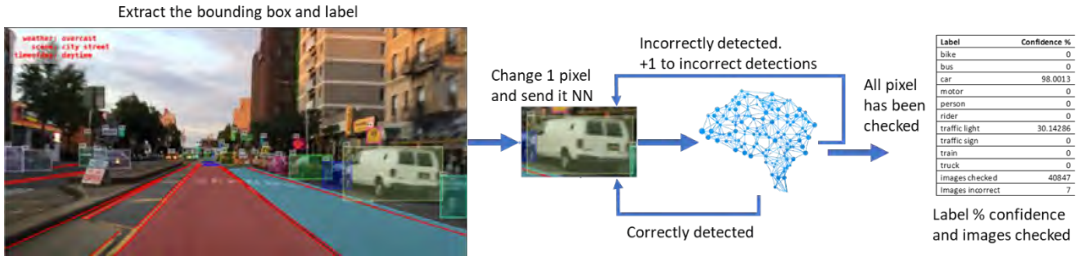


図 43 BDD 100k を用いた 1 ピクセル攻撃法

図 43 は、BDD100k 学習データセットで 1 ピクセル攻撃がどのように機能するかを示している（付録 C も参照）。まず、データセット中の各バウンディングボックスについて、そのボックスで区切られた画像を抽出する。次に、その画像の 1 ピクセルを変更する。最後に、変更した画像を検出モデルに送り、それでも正しく物体を認識できるかどうかを判断する。ピクセルの変更方法としては、ピクセルの RGB 色をその反対の RGB 色に変換することにした。

表 25 1 ピクセル攻撃の確信度

ラベル	NN による確信度
自転車	0
バス	42.86
自動車	62.59
バイク	21.68
人	0
ライダー	0
信号	0
標識	29.48
列車	0
トラック	95.15
検査した画像数	86390
認識失敗画像数	68327
失敗した割合(%)	79.09133



図 44 1 ピクセル攻撃した
トラックの画像例

この処理は、バウンディングボックスの各ピクセルに対して行う。毎回、画像内の 1 ピクセルだけを変更することに注意。画像内のすべてのピクセルを処理した後、この方式は変更したすべての画像をチェックした確信度ラベルと、チェックした画像の数をまとめる。例えば、300x200 ピクセルの画像に対して、この処理は検出モデルに 6 万枚の画像をチェックさせ、6 万枚の画像の結果を集計する。

図 43 のバウンディングボックスに対し、Yolov4 を検出モデルとして 1 ピクセル攻撃を適用した画像の確信度を同じ図の中に示した。これを見ると、この検出モデルは画像を**自動**

車または**信号**と認識している。しかし、**自動車**と認識した場合の確信度は 98%、**信号**と認識した場合は 30.14%の確信度に留まっている。この処理では、40,847 枚の画像をチェックし、そのうち 7 枚で誤検知した、つまり 0.017%しかノイズの影響を受けていない。この結果から、このバウンディングボックスはノイズの影響を受けにくく、モデルの学習に適していることがわかる。

別の例として、図 44 の画像と Yolov4 を検知モデルとした 1 ピクセル攻撃では、検知モデルはこれを**トラック**の他、**自動車**、**バス**、**バイク**、**標識**と認識することがある。表 25 に示すように、この画像を**自動車**と認識する確信度は 60%を超えている。これは、この画像が**トラック**と**自動車**の間のコーナーケースで、検出モデルの訓練が正しく行われなかったためである。また、この画像にノイズがある場合、このシステムは約 79%の確率で正しい認識に失敗する。この画像は、1 ピクセル攻撃によって、コーナーケースや訓練プロセスに適さないバウンディングボックスを検知できることが分かる、よい例となっている。

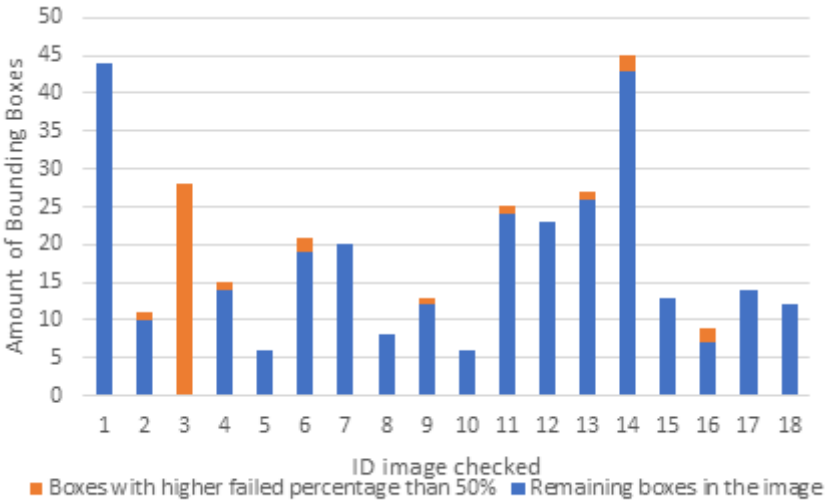


図 45 不正なバウンディングボックスの検出に
1 ピクセル攻撃を用いた ABCI による実験

1 ピクセル変更は、NN にノイズを発生させている画像や、コーナーケースを検出できる。しかし、特にデータセット内のすべてのバウンディングボックスを調べたい場合には、処理時間が長くなってしまう。例えば、次の実験では、ABCI サーバ（強力なサーバ機）を 8 時間使用した。その間に、1 ピクセル攻撃は 340 個のバウンディングボックスを含む 18 枚の画像をチェックすることができた。これは、1 時間あたり約 43 個のボックスをチェックしたことになる。訓練用データセット全体では、1,272,818 個のバウンディングボックスがある。つまり、BDD100k のトレーニングデータセット全体をチェックするのに約 29,600 時間（1233.33 日、3.38 年）かかるということである。さらに、このように不良バウンディングボックスを検出した後、モデルの再トレーニングを行うなどの対処が必要である。

図 45 は、BDD100k で 1 ピクセル攻撃を 8 時間行った ABCI 実験の結果である。この

結果から、この 18 枚の画像のバウンディングボックスのうち、11%が訓練処理に適さないことがわかる。さらに、画像 3 の結果を見ると、検出モデルにノイズしかもたらしていない。残念ながら、時間の制約により、BDD100k データセットのすべてのバウンディングボックスをチェックすることはできず、その結果、コーナーケースを除去した後に検出モデルを再トレーニングし、得られた精度を確認することができなかった。とはいえ、この問題を解決する方法が 2 つある。1 つは、アルゴリズムの改良で、例えば、並列化を採用して高速化したり、全ラベル同時ではなく、1 つのラベルのみに着目したりすることである。もう一つは、1 ピクセル変更と surprise adequacy を組合せる方法である。これは、surprise adequacy を用いてコーナーケースを求め、検出したコーナーケースを 1 ピクセル変更で分類するものである。この方法では、1 ピクセル変更は少数のバウンディングボックスをチェックするだけで、失敗率が最も高いコーナーケースを優先的に検出する。さらに、1 ピクセル変更と surprise adequacy を併用した検査で得られる確信度により、どのラベルがコーナーケースに近いかを判断することができる。

7.5.8 D-1: プログラムの信頼性

MLQM ガイドラインでは、**プログラムの信頼性**とは、基盤となる従来のソフトウェア（訓練プログラム、予測・推論プログラムなど）が正しく機能することを意味する。この概念には、アルゴリズムの正確さ、時間・メモリ資源の制約、ソフトウェアセキュリティなどのソフトウェア品質要求を含む。したがって、コンポーネントの健全性を保証するために、ソリューション設計者は以下のことを行う必要があり、これがここでのポイントである。

- アルゴリズムの正しさを確認する。
- オープンソースの実装を使う場合は、よく検証された確かなものを選択する。
- ライブラリの頻繁なバージョンアップに伴う不具合を解消する。
- テスト環境と実運用環境がどの程度一致しているかを確認する。

アルゴリズムの正しさ

アルゴリズムの正しさについて、まず、**正しさ**とは何かを理解する必要がある。この場合、アルゴリズムの正しさとは、そのアルゴリズムが 仕様に照らして期待通りの出力を正しく行い、かつ、エラーなく終了することである。これは**完全正当性**と呼ばれる。

例えば、本検討では、様々な目的のためにいくつかの追加アルゴリズムを設計した。すなわち、頑健性評価（1 ピクセル変更）、データアノテーション（アノテーションツール）、データラベル変換（nuScene ラベルから BDD100k ラベルへの変換）である。これらのアルゴリズムの正しさを証明する唯一の方法は、考えられるすべての入力を使ってテストし、その出力を分析することである。アルゴリズムは、エラーや警告を出すことなく、実際に期待通りの出力を生成した。

オープンソース要素の健全性

この AI の開発には、Python 言語を使用した。Python は様々なオープンソースのパッケージを使用しており、これらは互いにバージョン互換性があることが望ましい。そのため、開発者は、使用するパッケージとそのバージョンの一覧を提供する必要がある。

表 26 本検討で使用したオープンソースパッケージの一覧

プログラミング言語	バージョン
Python	3.8.3
パッケージ	バージョン
NumPy	1.18.5
TensorFlow	2.3.1
PyTorch	1.6.0
SciPy	1.5.2
Pandas	1.1.0
Matplotlib	3.3.0

検出モデルごとにライブラリや依存関係が異なるために起こりうるあらゆる問題を回避するために、使用する各検出モデルのライブラリや依存対象をすべて含む docker イメージを作成して用いた。

訓練および運用環境におけるハードウェアの信頼性

コンポーネントの健全性を検証するためには、使用するハードウェアが非常に重要である。物体認識や画像分類のモデルは複雑なため、訓練と推論には GPU を使うことが望ましい。この検討の現段階では、推論のみを行う場合は GPU を使用せずに docker イメージを使用することが可能である。しかし、検知モデルを再訓練したいなら、GPU を搭載したマシンやサーバーを使用する方がよい。

この docker は、ハードウェア仕様の異なる 5 種類のマシンやサーバーでテストした。その結果、docker は新しいハードウェアに問題なく適応して動作することが確認できた。

メモリ使用量の健全性

プログラムが運用環境でメモリ不足に陥らないように、学習時と推論時の最大メモリ使用量を適切に記録し、評価する必要がある。これらの記録に基づいて、運用環境におけるデバイスの効率的なメモリ割り当てを実現することができる。

- **モデルのアーキテクチャと重み**　訓練済み AI ネットワークとその重みを保存するために使うファイル形式の 1 つに HDF(Hierarchical Data Format)ファイル(.h5)がある。これを用いると、例えば、YOLOv4 の訓練済みネットワークは 490,961 のパラメータを持ち、ハードディスクに約 52.43MB の空きを必要とする。
- **ソースコード**　プログラミング言語によって書かれたさまざまなアルゴリズムは、装置のワークフローの一部である。これらのコードは通常、データや訓練済みの重みに比べて記録メディアにあまり場所を取らない。
- **入力データ**　訓練用およびバリデーション用データセットに必要なメモリは、サンプル数や画像の解像度によって非常に大きくなる可能性がある。この検討では、BDD100k 動画データセットから抽出した画像の解像度はすべて 1280×720 ピクセ

ルだった。訓練用データセット（画像 7 万枚）、バリデーション用データセット（画像 1 万枚）、テスト用データセット（画像 2 万枚）のメモリ使用量はそれぞれ 3.77GB、553MB、1.07GB である。入力データの解像度やサイズは、運用環境においてカメラの設定やその他の理由で変化する可能性があるため、訓練時および推論時の入力データのメモリ使用量を記録しておく必要がある。

- **演算ユニットの仕様**　訓練時と推論時の両方で必要な最小限の RAM と GPU を調べ、記録する必要がある。

訓練時間や推論時間の効率

実生活に適用した ML モデルでは、時間は非常に重要な問題である。自動運転車の場合、命に関わるような事例を正しく認識した後、一瞬のうちに重要な決断を下さなければならないかもしれない。したがって、推論時間が短いほど、AI 製品の信頼性は高くなる。推論時間は、すべての最終候補モデルについて記録し、推論処理にかかる時間が十分に短いかどうかを評価する必要がある。

また、訓練の時間を記録し、訓練プログラムのどこかで不必要な時間ロスが発生していないかもチェックする必要がある。開発工程では、より高速なアルゴリズムを使用することが望ましい。

7.5.9 E-1: 運用時品質の維持性

MLQM ガイドラインの 6.9 項によると、**運用時品質の維持性**とは、運用開始時に実現していた内部品質が運用中も維持されることである。

運用時には、開発したシステムを更新したり、新しいデータを訓練用およびバリデーション用データセットに組み込んでモデルを改善する。そのため、機械学習を用いたシステムや機械学習コンポーネントの動作を運用中に継続的に監視することが求められる。

システムを更新し続けるには、2 つの運用パターンがある。1 つは、開発フェーズに戻り、新しい変更を加えた後に全体のプロセスを修正し、システムを再展開する方法である。もうひとつは、必要なソフトウェア部品を更新してシステムを更新し続ける方法である。そのためには、運用中にリアルタイムで必要なコンポーネントを更新する必要がある。2 番目の方法の方が速いが、不適切な依存対象を受け取るリスクが高い。

ここでのポイントは以下の通り。

- 精度のモニタリング
- モデル出力のモニタリング、入力データのモニタリング
- KPI のモニタリング

精度のモニタリング

精度のモニタリングとは、検知モデルの精度を監視し、精度向上に応じて情報を更新する手法である。本検討では、検知モデルの精度を、mAP (mean Average Precision)を用いて測

定している。ここまでの節では、mAP を監視し、検知モデルの改良に利用する方法について述べた。これらの改善は、データセットの画像を改善したり、モデルを再訓練したりすることによって行った。

モデル出力と入力データのモニタリング

モデル出力モニタリングと**入力データモニタリング**は、それぞれ訓練済みの機械学習モデルによる推論結果の監視と、その入力データの監視を意味する。ここではどちらにも 1 ピクセル変更と surprise adequacy という 2 つの異なる仕掛けを使用した。これらの仕掛けにより、訓練処理にノイズを加えるコーナーケースを検出、修正、削除して、訓練用データセットを更新することができた。

KPI のモニタリング

KPI のモニタリングは、検出モデルを KPI の観点から監視することに重点を置いている。KPI（Key Performance Indicator）とは、機械学習ベースのシステムを通じて機械学習コンポーネントからの出力が達成すべき機能要求の達成度を定量化する指標である。つまり、KPI とは、プロジェクトがいかに効率的に主要な目的を達成しているかを示す測定可能な値である。KPI の定義は厄介である。ここで重要なのは Key という単語で、なぜなら、すべての KPI はプロジェクトの達成度が示せる具体的な効果に関連するものでなければならないからである。KPI は、重要なまたは中核的なビジネス目標に従って定義する必要がある。

KPI を定義するためには、以下の質問に答える必要がある。

- あなたが望む効果は何か？
- なぜ、この効果が重要なのか？
- 進捗状況をどのように把握するつもりか？
- どのように効果に影響を与えることができるのか？
- 効果を達成したことはどうすると分かるか？
- 効果に向けての進捗状況をどの程度の頻度で確認するか？

この検討では、自動運転をテーマとしたので、自動運転を行う機械学習を定義するにあたり、安全性レベルの標準的定義を使うことにした。すなわち、ASIL (Automotive Safety Integrity Level) D である。ASIL D は、誤作動の際に生命を脅かす、あるいは致命的な傷害を負う可能性が高いことを表し、関連する安全性目標が十分で、達成されていることを最高レベルで保証する必要があることを示している。

今回の検討では、BDD100k から以下の属性を使用した。

天気：雨、雪、晴れ、曇り、一部曇り、霧、未定義

場面：トンネル、住宅地、駐車場、市街路、ガソリンスタンド、高速道路、未定義

時間帯：昼、夜、明け方または夕暮れ、未定義

信号の色：赤、黄、青、なし

歩行者の有無：有、無

横断歩道の有無：有、無

明度：非常に高い、高い、中間、低い、非常に低い

合計 7 つの属性と 31 の属性値がある。

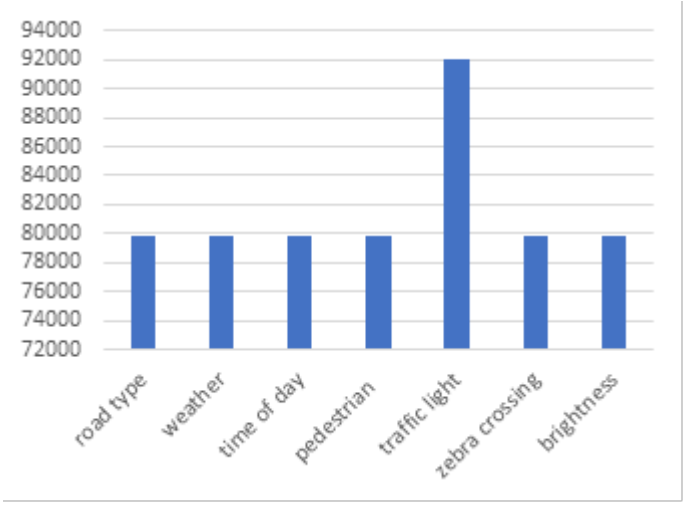


図 47 BDD 100k における属性ごとの画像数

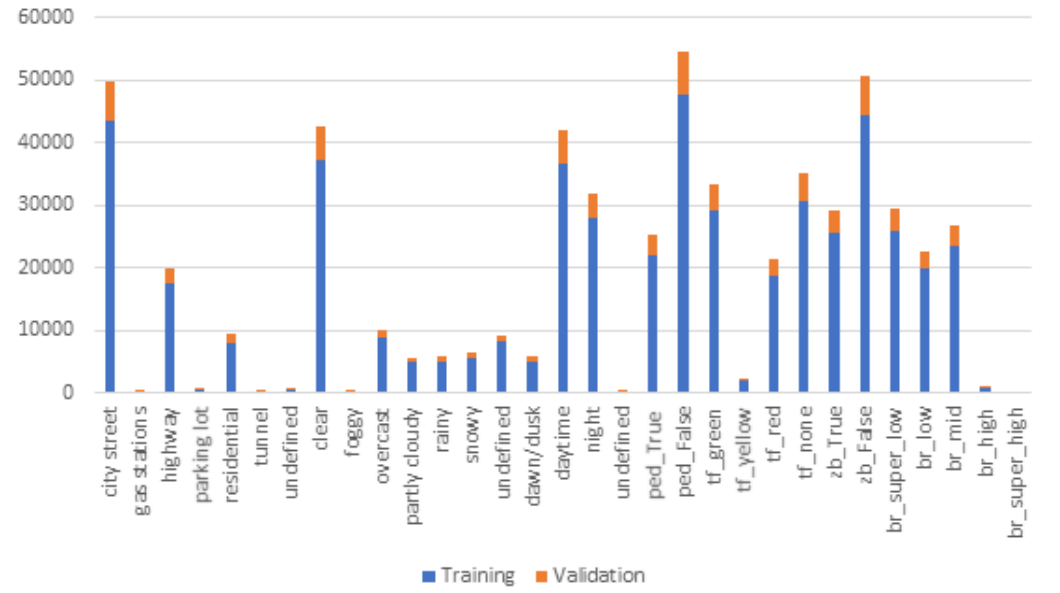


図 46 BDD 100k の属性値および
訓練・検証データごとの画像分布

図 46 と図 47 を見るとわかる通り、**市街路**（49628 枚）や**ガソリンスタンド**（34 枚）など、出現頻度が他の属性より高いものも低いものもある。しかし、KPI を定義するためには、個々の属性値を使うわけにはいかない。KPI を定義するために注目すべき状況を見つけたいのである。そのため、属性値を 2 つずつ組合せることにする。そうすると、375 通り

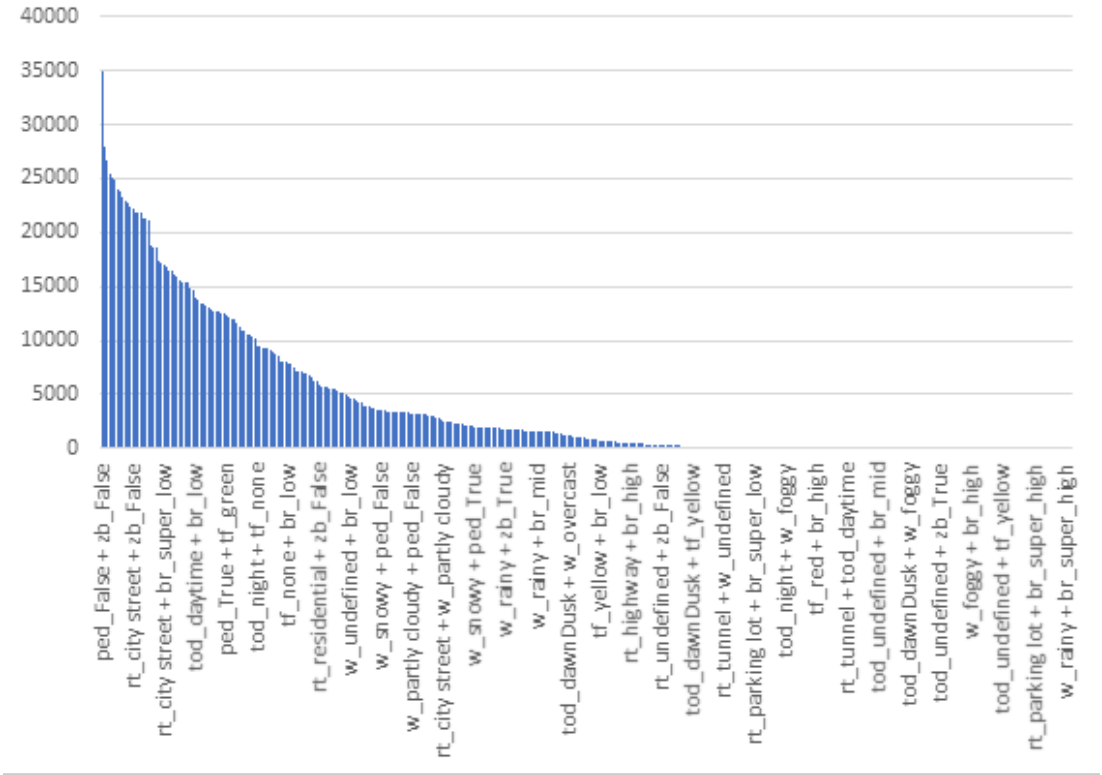


図 48 BDD 100k の 2 属性値組合せごとの画像量

の組合せになる。

図 48 は見づらいので、最初の 30 個の組合せに着目してみよう。

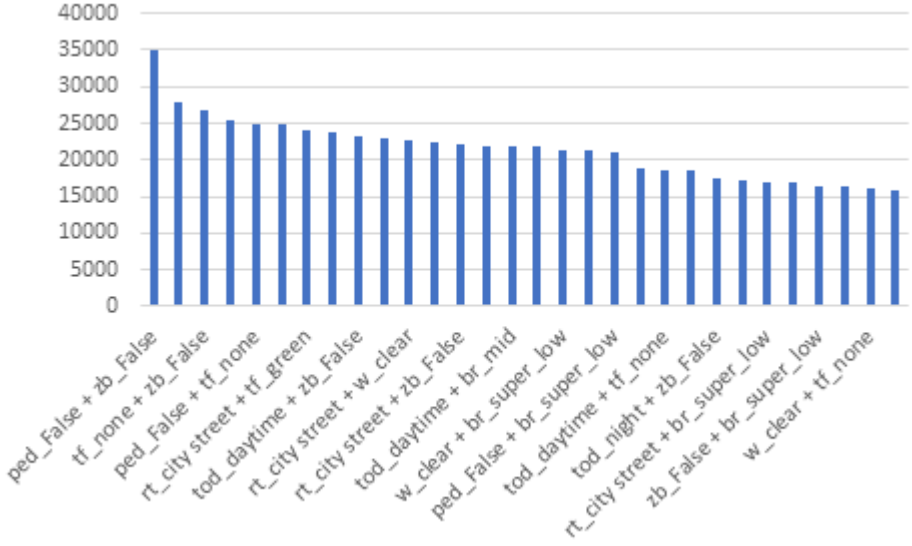


図 49 BDD 100k の 2 属性値の組合せごとの画像量上位 30 位

すべての組合せに十分な画像がないため、KPI の定義にすべての組合せを使用することはできない。このため、組合せごとの画像量と ASIL D を考慮して、**道路種別、天気、時間帯**にのみ焦点を当てることにした。この 3 つは、画像の見やすさに関して、他より影響が大

きい。その結果、他の項目よりもASIL Dとの関連性が高い。実のところ、**明度**を含めた方がよいかもしれない。必要なら、今後の課題とする。

また、属性値**未定義**は実態がないためカウントしない。すると、63通りの組合せが残る。

表27 道路種別、天気、時間帯の2属性値組合せ画像の枚数

ID	組合せ	枚数	ID	組合せ	枚数
1	時間帯_夜＋天気_晴れ	22884	33	道路種別_住宅地＋天気_一部曇り	580
2	道路種別_市街路＋天気_晴れ	22750	34	時間帯_夜明け夕暮れ＋天気_一部曇り	570
3	道路種別_市街路＋時間帯_昼	21811	35	道路種別_住宅地＋天気_雨	487
4	道路種別_市街路＋時間帯_夜	18748	36	時間帯_明け方夕暮れ＋天気_雪	436
5	時間帯_昼＋天気_晴れ	12454	37	時間帯_明け方夕暮れ＋天気_雨	328
6	道路種別_高速道路＋天気_晴れ	10422	38	道路種別_駐車場＋時間帯_昼	228
7	道路種別_高速道路＋時間帯_昼	8905	39	道路種別_駐車場＋天気_晴れ	169
8	時間帯_昼＋天気_曇り	7551	40	道路種別_駐車場＋時間帯_夜	94
9	道路種別_高速道路＋時間帯_夜	7025	41	時間帯_夜＋天気_曇り	72
10	道路種別_住宅地＋時間帯_昼	5658	42	時間帯_夜＋天気_霧	67
11	道路種別_市街路＋天気_曇り	5121	43	道路種別_駐車場＋天気_曇り	62
12	時間帯_昼＋天気_一部曇り	4262	44	道路種別_市街路＋天気_霧	61
13	道路種別_市街路＋天気_雪	3996	45	天気_夜＋天気_一部曇り	49
14	道路種別_住宅地＋天気_晴れ	3800	46	時間帯_昼＋天気_霧	48
15	道路種別_市街路＋天気_雨	3395	47	道路種別_高速道路＋天気_霧	41
16	道路種別_市街路＋時間帯_夜明け夕暮れ	2950	48	道路種別_駐車場＋天気_一部曇り	33
17	時間帯_昼＋天気_雪	2862	49	道路種別_駐車場＋天気_雪	32
18	道路種別_市街路＋天気_一部曇り	2561	50	道路種別_トンネル＋時間帯_昼	32
19	時間帯_昼＋天気_雨	2522	51	道路種別_駐車場＋時間帯_夜明け夕暮れ	30
20	道路種別_高速道路＋天気_曇り	2336	52	道路種別_住宅地＋天気_霧	27
21	時間帯_夜＋天気_雪	2249	53	道路種別_駐車場＋天気_雨	21
22	時間帯_夜＋天気_雨	2208	54	道路種別_トンネル＋時間帯_夜	20
23	時間帯_夜明け夕暮れ＋天気_晴れ	2004	55	時間帯_夜明け夕暮れ＋天気_霧	15
24	道路種別_住宅地＋時間帯_夜	1813	56	道路種別_トンネル＋天気_晴れ	8
25	道路種別_高速道路＋天気_一部曇り	1705	57	道路種別_トンネル＋天気_雨	7
26	道路種別_高速道路＋時間帯_夜明け夕暮れ	1439	58	道路種別_駐車場＋天気_霧	1
27	道路種別_住宅地＋天気_曇り	1239	59	道路種別_トンネル＋時間帯_夜明け夕暮れ	1
28	時間帯_夜明け夕暮れ＋天気_曇り	1147	60	道路種別_トンネル＋天気_霧	0
29	道路種別_高速道路＋天気_雨	1105	61	道路種別_トンネル＋天気_曇り	0
30	道路種別_住宅地＋天気_雪	795	62	道路種別_トンネル＋天気_一部曇り	0

31	道路種別_高速道路＋天気_雪	707	63	道路種別_トンネル＋天気_雪	0
32	道路種別_住宅地＋時間帯_夜明け夕暮れ	599			

中には知り得ない状況や非現実的な状況がある。例えば、トンネルの中で天気が一部曇り（62 番）かどうかは分からない、など。一方、現実的な状況でも、その組合せの精度を確かめるには画像数が足りない場合がある。ここで画像 100 枚を閾値として使っている。例えば、**道路種別**（**住宅地**）＋**天気**（霧）［ID 52］には 27 枚の画像があり、ラベルごとの mAP は以下の通りである。

表 28 属性値住宅地と霧の組合せのラベル精度の例

組合せ	自転車	バス	自動車	バイク	人	ライダー	信号	標識	列車	トラック	平均値	IoU
住宅地＋霧	0	0	57.61	0	0	0	0	100	0	0	56.98	15.76

このラベル精度から、**住宅地**＋**霧**の組合せでは、住宅地にある物体（**人**、**自転車**など）を検出するためにはより多くの画像が必要であることはわかる。しかし、正しい精度を得るための十分な画像がないため、このラベル精度自体は正しくない。

残りの属性に着目すると、任意の天気や時間帯における以下の**道路種別**が KPI で考慮するべき事項となる。

- **市街路**
- **高速道路**
- **住宅地**
- **駐車場**
- **ガソリンスタンド**

トンネルについては、この属性値は独立しているので、単独で一つの KPI として確認する必要がある。残りの**時間帯**と**天気**の組合せについては、すべて有効である。その結果、**55 種類の KPI を作成することができる。**

KPI の例：住宅地＋霧

図 50 は、BDD100k データセットにおける住宅地と霧の属性値の例である。この例の画像を含め、属性値として住宅地と霧を持つすべての画像を調査した。必要なデータを入手したので、この組合せに関連する KPI を定義する準備ができた。



図 50 属性値住宅地＋霧の例

- あなたが望む効果は何か？
  - **列車**と**トラック**以外のすべてのラベルを最低 60%で検出する。この 2 つは**住宅地**では使用できないので、重要ではない。
- なぜ、この効果が重要なのか？
  - ASIL D を達成し、誤検知による健康被害を回避することが重要だから。
- 進捗状況をどのように把握するつもりか？
  - 物体検知の**精度**と surprise adequacy を使用する。
- どのように効果に影響を与えることができるのか？
  - 精度の低い特定のラベルや属性の画像を追加すること。そのために、追加データを使用する。今回の場合、追加データを使っても、**住宅**と**霧**の属性を共有する画像はあと 21 枚しか見つからなかった。データ拡張の仕掛けを強制的に働かせ、これらの画像をより多く取得することが可能である。
  - さらに、画像処理アプリケーションを使ってデータセット内の現在の画像を修正し、修正後の画像をデータセットに追加すれば、より多くの画像を含めることができる。
  - また、何も見えない画像などの不良画像は、精度が悪すぎる（30%未満）上に、訓練プロセスにノイズを加えるだけなので、削除してもよい。

- 個々の検知モデルを、**住宅地**や**霧**の中でのみ動作するように訓練する。

- 効果を達成したことはどうすると分かるか？
  - 画像の拡張が実現した場合。
  - すべてのラベル、または少なくともほぼすべてのラベルで精度が 60%より良い場合。
  - 訓練した検知モデルを nuScene などの他のデータセットの画像を用いて確認する。
- 効果に向けての進捗状況をどのくらいの頻度で確認するか？
  - 新しい KPI が出るたびにチェックする。
  - データセット外の**住宅地＋霧**の属性を持つ画像群を新たに推論に用い、いずれかのラベル（**トラック**と**列車**を除く）の精度が 60%より低かった場合。

これらの質問に答えたことで、**住宅地**と**霧**の組合せの KPI が定義され、BDD100k で頑健性を達成したい人に利用可能になった。将来的には、定義したすべての KPI を他のデータセットでテストし、その性能を確認する予定である。また、必要に応じ KPI も更新する予定である。

7.6 用語集

7 章に示した検証手順の記述にある専門用語の一覧を以下に示す。ここに記載されていない定義については、MLQM ガイドラインの 2.3 節を確認するとよいかもしれない。

物体検知

画像中の物体の存在と位置を特定する作業。

アノテーション

画像中の対象物の面積、位置、種類（クラス）を特定する正解データで、通常、対象物を囲む矩形（バウンディングボックス）とラベルで与えられる。

データセット

機械学習アルゴリズムの訓練とバリデーションに使用される画像とアノテーションのセット。

訓練

データセットのサブセットから画像とアノテーションを繰り返し与えてモデルを学習させ、そのパラメータを修正して結果を向上させるプロセス。

バリデーション

データセットの中の、訓練時に見ていない部分を使って、モデルが物体を検出する能力を評価するプロセス。

IoU（Intersection over Union）

IoU は Jaccard 指数とも呼ばれ、2 つの集合の積集合の大きさと和集合の大きさの比である。物体検知の文脈では、人間がアノテーションで示した、意味のある領域に対する、予測した領域の精度を、位置と大きさに関して測定するのに使う。

混同行列

真陽性（TP、対応する正解物体のある、検知したという予測）、偽陽性（FP、対応する正解物体のない、検知したという予測）、真陰性(TN、対応する正解物体がない、存在しないという予測)、および偽陰性（FN、存在する正解物体に対する、存在しないという予測）の数を報告する行列である。

適合率と再現率

機械学習アルゴリズムの物体検知精度を示す 2 つの指標。混同行列から定義される。ここで、適合率は $\mathrm{TP}/(\mathrm{TP}+\mathrm{F}P)$ であり 再現率は $\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$である。

mAP (mean Average Precision)

物体検知性能の指標。IoU の閾値が与えられた場合、評価データセット内の物体の全クラスについて、閾値以上の IoU を持つ予測について計算された平均精度値の平均として計算される。すべての検出の平均精度は、適合率対再現率曲線の下での面積である。

FPS（Frames Per Second）

機械学習アルゴリズムが 1 秒間に評価できる画像の枚数。

KPI（Key Performance Indicator）

機械学習アルゴリズムが達成すべき特定の目標 mAP のこと。

候補モデル

自動運転システムの物体検知方法として検討対象とする機械学習アルゴリズム。

最終候補モデル

機械学習のアルゴリズムで、候補モデル群から開発に使うために選択したもの。

8 金属鋳物の外観検査

8.1 ビジネス要件

8.1.1 背景

製造工程では、さまざまな理由により、新たに生産された製品に欠陥が生じることがある。例えば、金属鋳造製品の表面には、小さな穴や亀裂、割れなどの欠陥が存在する。これらの欠陥は、鋳造製品の外観にとどまらない様々な品質に影響を及ぼす。そのため、外観検査による欠陥検出は、製造業において非常に重要なテーマとなっている。このような不良品を検出するために、あらゆる産業界には品質検査部門が存在する。しかし、この検査工程は通常、目視確認で行われていることが大きな問題である。非常に時間のかかる作業であり、検査者の感覚のばらつきやゆれもあるため、100%正確とは言い切れない。もし、不良品率が顧客の要求を満たさなければ、会社に大きな損失をもたらす可能性がある。そのため、機械学習による自動欠陥検査への期待が高まっている。

8.1.2 目的・目標

- 新しく生産されるすべての金属鋳造製品の迅速な自動欠陥検出を実現。
- 検出された不具合の種類を分類し、機器のメンテナンスに役立てる。
- 鋳造欠陥の検出において、従来の手作業による検出より高い精度を実現し、品質検出における人的・経済的コストを削減。
- 夜間や強い反射のある時など、人が働きにくい環境下での製品不良検出を実現し、製造工程の安全性を向上させる。

8.1.3 AI システムのステークホルダー

- 製造事業者
- 鋳造製品ユーザー

8.1.4 ステークホルダーの初期要求

- 製造事業者や鋳造製品ユーザーは、このシステムによって不良品の大半を捕捉し、不良品にかかるコストを削減し、収益を上げることを期待している。
- 製造事業者はさらに、この一連の鋳造製品の品質を把握できるよう、AIシステムが鋳造製品の試験報告書を生成することも期待している。

8.1.5 ビジネス要件の詳細

前提条件

- 金属鋳造製品の表面から取り込んだ画像で、高精度なAIモデルを訓練することができる。

依存事項

- 運用時には、鋳造品の組立ラインの上に撮像用のカメラを設置し、トップビューで撮像すること。
- 適切な角度で撮影されていること
- AIシステムに入力される画像には、埃によるノイズがないこと。

制約事項

- 金銭的制約：AIシステムの開発・保守にかかる諸経費は、目視確認による欠陥検出にかかる費用よりも安いこと。

機能要件

- 鋳造品の不良を認識できること。

非機能要件

- 精度が目標とする品質標準を満足するものであること。
- 認識プロセスは、誤認識率やヒット率など、様々な誤差基準を満たし、また、目標とする品質標準が定める閾値に達すること。
- 所定の範囲内の様々な解像度の画像に対して頑健なシステムであること。
- 様々なメーカーのカメラで撮影された画像に対し頑健なシステムであること。

考慮しない事項

- 鋳造品の表面にある欠陥の位置を特定することは、当面、システムの対象外とする。

リスクと安全に関する懸念

- AIシステムが不安定だと、不適格な鋳造品が出荷されて、顧客の期待が満たされない恐れがある。

8.1.6 外部品質に関する要求事項

開発する AI ソフトウェアに期待される品質要求レベルを、3 つの主要な外部品質について以下に示す。

安全性

- このシステムに関連して傷害の懸念はないため、その点に関する要件はない。

- 不良品を多数出荷すると、業績に多大な悪影響が及ぶ恐れがある。多数の不良品の出荷を防ぐため、欠陥を見逃す割合は十分低い必要がある。

パフォーマンス

- このシステムは、鋳造分野で一般的に使用される工業規格を満たす必要がある。そのため、欠陥判定は、当該工業規格にそっている必要がある。
- 適格品を欠陥と誤判断する割合が高いと、歩留まりが下がったり、人による再確認が必要になったりし、自動化の狙いが果たせない。そのため、適格品を誤判断する割合が十分低い必要がある。

公平性

- このシステムに関連して公平性の懸念はないため、その点に関する要件はない。

8.1.7 達成すべき外部品質特性レベルの特定

表 29 実現すべき外部品質特性レベル

外部品質	補足説明	想定される深刻度	実現すべきレベル
安全性	人的リスクに対する AI 安全レベル	物理的なダメージは想定されない	AISL 0
	経済的リスクに対する AI 安全レベル	軽微な利益損失、人による監視で回避可能	AISL 0.2
パフォーマンス	一般的 AI パフォーマンスレベル	KPI は事前に特定されるが、各 KPI の閾値は他の要因によって変動する可能性があり、ベストエフォートで提供される	AIPL 1
公平性	製品・サービスの公正さに対する明確な要求はない		AIFL 0

8.2 品質マネジメントの手順

外観検査には、繊維生地検査、製品検査、自動車の欠陥検査など、さまざまな用途がある。このリファレンスガイドでは、特定用途、すなわち、金属鋳造製品を取り上げ、製品の表面欠陥を検出することに焦点を当てる。データセットとしては、鋳造欠陥に関するデータ[21]を用いる。鋳造欠陥とは、金属鋳造プロセスにおける望ましくないむらのことである。鋳造欠陥には、鋳巣、ピンホール、バリ、引け巣、注湯欠陥、冶金的欠陥など、さまざまな

種類がある。選択したデータセットには、合計 7, 348 枚の画像データが含まれている。これらの画像は全て、水中ポンプインペラーの上面図であり、サイズは (300*300) ピクセル、グレースケール画像である。このデータセットは、学習用とテスト用に 2 つのフォルダに分割されている。train と test の両フォルダには、以下のように def-front と ok-front のサブフォルダがある。

train: def-front 3758 枚、ok-front 2875 枚

test: def-front 453 枚、ok-front 262 枚

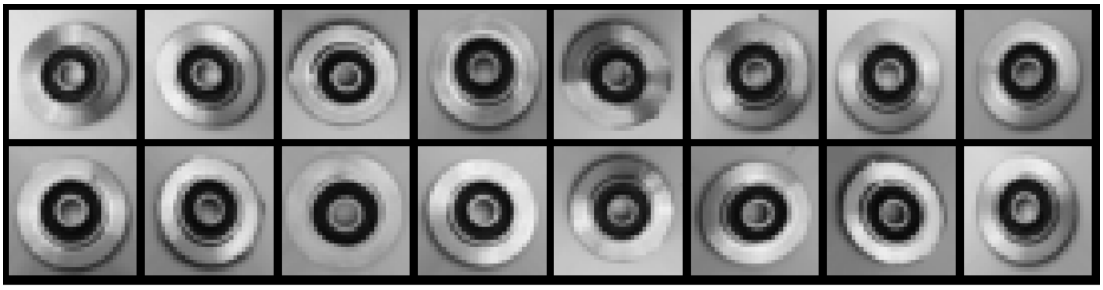


図 51 鋳造欠陥データの例

8.2.1 A-1: 問題領域分析の十分性

問題領域分析の十分性に関するおおまかな定義を踏まえて、外観検査アプリケーションでは、要件をいくつかの側面に具体化することができる。例えば、アルゴリズムや構造の観点からは、外観検査問題に適した機械学習アルゴリズム・構造を把握する必要があり、構築した外観検査 AI システムは実世界で実行する必要がある。データの観点では、外観検査の問題領域の定義、想定されるすべての状況に対するデータの網羅性、高リスクのケースの定義が要件となる。また、実行の観点では、PoC の段階で KPI を設定することが必要である。

AI モデルの要件

鋳造欠陥データを外観検査する場合、関連する問題として、欠陥の検出と分類がある。今回のデータセットには、欠陥と正常の 2 種類の画像しか含まれていないため、標準的な 2 値分類問題となる。したがって、CNN [22]、MLP [23]、および ResNet [13]のような一般的な分類モデルが適用できる。これらの AI モデルには、畳み込み層、Max プーリング層、全結合層、意思決定層（分類層）などが含まれる。また、構築された AI モデルの性能を評価するために、適合率、再現率、正解率などの KPI も有用である。

データ型の要件

さらに、機械学習用データの種類にも重要な要件がある。1 つ目は、AI モデリングのための入力の種類である。問題領域記述のための属性とは異なり、AI モデリングのための入力は具体的な値や型（離散的か連続的か）を持つ必要がある。鋳造欠陥の外観検査システムの場合、入力は 1 チャンネルのグレーフォーマット画像で、ターゲットは欠陥の有無を表

す 2 値のクラスラベルであることが必要である。

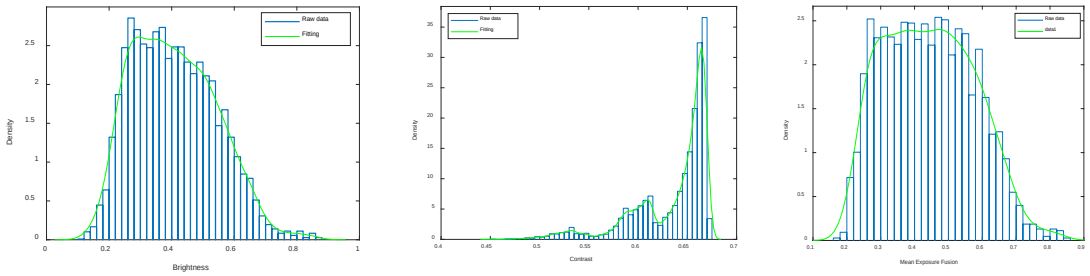
2 つ目は、問題領域記述のための属性の種類である。この要件はより一般的であり、離散的な値と連続的な値の両方が許容される。通常、強い光量、弱い反射など、問題領域の自然な特性を表現するには、離散的な値がより適している。

関連する属性と属性値を特定する

与えられた鋳造欠陥データに基づき、具体的な AI システムの問題領域を定義するのに有効ないくつかの関連する属性を特定することができた。本節で扱うモデリングプロセスでは、入力にはある鋳造製品のトップビューの画像のみを含む。それを踏まえると、定義できるのは画像に関する属性である。例として、鋳造欠陥のデータに関し、以下のように属性要件を決める。

- タイプ：欠陥あり、欠陥なし
- 輝度：強、中、弱
- コントラスト：強、中、弱
- 露光：強、中、弱

欠陥あり画像と欠陥なし画像の 2 種類しかない鋳造欠陥データについても、ここでは輝度、コントラストや、さらに露光 [24]などの属性でその問題領域を記述することができる。これらの属性の分布を図 52 に示す。



(a) 輝度　　(b) コントラスト　　(c) 露光

図 52 鋳造欠陥データの属性の分布

図 52 から分かる通り 3 つの属性は連続的に分布しているので、ここでの外観検査の要求分析では、次の表の通り定義域を決めておく。また、外観検査の問題を記述する際には、適切な閾値を決めて、強、中、弱のケースに分けることができる。

表 30 与えられた属性の定義域

属性	輝度	コントラスト	露光
定義域	0~1	0.4~0.7	0.1~0.9

POC 段階での KPI 要件

さらに、POC（Proof of Concept）トライアルの段階では、通常、開発者は構築した機械学習ベースの外観検査システムに対して、いくつかの KPI 要件を提案することができる。これらの KPI 要件は、モデル、データ、パフォーマンスなど、さまざまな観点から作成す

ることができる。

8.2.2 A-2: データ設計の十分性

外観検査アプリケーションにおいて、**データ設計の十分性**という内部品質には 3 つの側面が関わる。1）前節で述べた外観検査に基づく組合せ設計。ここで言う組合せは属性の組合せのことで、これにより、品質保証のためのケース（状況）の分割の仕方を決める。2）この品質にはケースの健全性も含まれ、これは各ケースのデータサンプルの量に直接関係する。サンプル数が少ないケースは、外観検査では不健全と判断される可能性がある。3）この品質は高リスクなケースにも関わる。例えば、特殊な状況下でのデータサンプルは、欠陥の誤認識や誤検出につながる可能性がある。

属性の組合せとケース分割

データ設計の十分性で最初に検討するのは、属性の組合せに基づくデータケースの設計である。ご存知のように、外観検査アプリケーションの問題領域は、いくつかの記述属性に基づいて構成されている。しかし、各属性の値がすべて妥当であるとは限らず、また、2 つの属性値の組合せは、個々の属性値が各々の属性領域には妥当であっても、組合せとしては妥当でない場合もある。このため、この段階では、ケースの分割や属性の組合せに関する分析が必要となる。

例として、実験用の鋳造欠陥データについて、AI 品質保証のために、問題領域設計段階での属性分布に従って、問題領域でのケース分割を設計してみる。

各属性（明るさ、コントラスト、露光）の問題領域を 10 個のサブケースに分割するとする。属性の組合せが異なるケースをまとめると、以下のようになる。

表 31 異なる数の属性の組合せによるケース分け

	件数
属性 1 つの組合せ	30 件
属性 2 つの組合せ	300 件
属性 3 つの組合せ	1000 件

コーナーケース/高リスクケース

属性の組合せとケース分割の設計の結果、いくつかのデータケースが得られる。このケース分割は、問題領域の分割を表しており、より詳細にデータを記述している。また、ケース分割に基づいて、さらにリスクの高いデータ領域や不健全なデータ領域、すなわち不健全ケースを発見することができる。

不健全ケース／高リスクケース

ここでいう不健全なケースとは、現実世界の外観検査では存在しないはずのケースのことである。例えば、**明るさ**属性の値を領域 [0, 1] に正規化すると、完全に暗い（brightness=0）、または完全に明るい（brightness=1）欠陥画像があり得る。このような画

像は不健全である。また、2 つの属性の組合せでも、暗すぎたり明るすぎたりする場合は、強いコントラストを持つことができないので、このような場合も不健全と見なされる。

高リスクケースは、不健全ケースとは異なり、外観検査が可能な現実世界に存在しうるものである。例えば、**鋳巣**の欠陥は実在し、見逃して出荷すると問題となるが、外観検査で検知するのは困難である。したがってこれは、外観検査の対象からは除外せざるを得ない。

コーナーケースデータ検出

従来のソフトウェアの実行と同様に、AI システムのテストにおいても、コーナーケースのデータを処理する際に危険な状態が発生する可能性があり、一般的に不適切で予期せぬ動作を引き起こす可能性がある [25]。例えば、深層学習(DL)ベースの自動運転システムが雨天や強い反射のコーナーケースを処理する場合、誤った判断がなされ、人命や財産の損失をもたらす事故につながる可能性がある [26]。したがって、コーナーケースのサンプルを検出することは、AI テストにおいて重要である。上記のコーナーケースの説明から、コーナーケース集合 [27]を以下のように定義することができる。

$$\boldsymbol{Corner\ case\ set}: \{\boldsymbol{x}|\boldsymbol{DL}(\boldsymbol{x}+\boldsymbol{perturbation}) \neq \boldsymbol{label}(\boldsymbol{x})\} \quad (1)$$

ここで、x はコーナーケースのサンプル、その真のラベルは $label(x)$、$DL(*)$は与えられた DL モデルに基づく出力クラスとする。この定義により、コーナーケースデータ x に、$|perturbation| < \varepsilon$ で $\varepsilon > 0$が小さな値であるような、小さな摂動を加えた場合、DL システムが認識するクラスは、その真のラベルとは異なるものとなる。このように、コーナーケース集合には、不適切で予想外の振る舞いをするデータサンプル、例えば、境界近くの敵対的データや誤って分類されたデータ（外れ値）などを含む（図 53）。

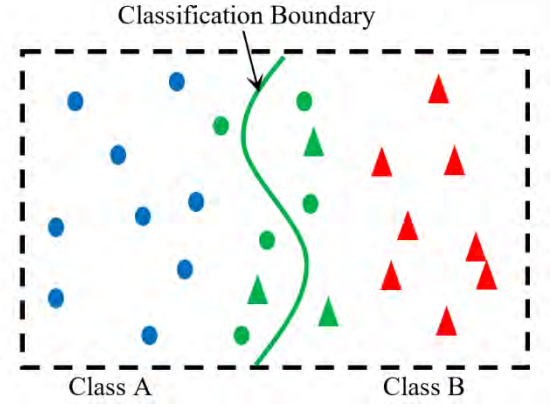


図 53 コーナー事例の模式図

2 つのクラスのデータを青と赤で色分けしている。コーナーケースのデータは、緑色で表示した。誤って分類されたデータや分類境界付近のデータなど、予期せぬ認識を起こしやすいデータが含まれる。

8.2.3 B-1: データセットの被覆性

機械学習による外観検査システムを構築する際、データの被覆性には、大域的被覆性と局所的被覆性という 2 つの視点がある。大域的被覆性は、主にデータの多様性を考慮したデータセット設計のことであり、例えば、訓練データの被覆性やテストデータの被覆性などが挙げられる。一般的な被覆性の定義によれば、訓練データの被覆性は、訓練用の問題領域のデータが十分にあり、外観検査の過程でデータ不足による不適切な学習動作が発生しないことを保証することを目的としている。テストデータの被覆性の目的は、構築した外観検査システムの問題領域における挙動を可能な限り完全に評価することである。局所的被覆性は、設計された各ケースにおけるデータ分布を調査することである。局所的被覆性の計算方法には、ケースの被覆性、属性に基づく被覆性、ニューロンベースの被覆性、Surprise Coverage など、様々な方法がある。

ケースの被覆性

被覆性の定義に直接基づいて、データ量から単純な被覆性メトリクスが容易に定義できる。なぜなら、データの被覆性はもともと、問題領域におけるデータ量と多様性を記述するものだからである。ケース分割、すなわち、外観検査の問題領域全体を、属性の組合せで細分化した小さなケースに分割することで、小さなケースに属性の多様性と属性値の多様性を反映させることができる。そして、データ量やデータ比率により、被覆性を容易に表現できる。

属性に基づく被覆性

また、問題領域で使用される属性から別の被覆性メトリクスを定義することもできる。この場合、属性の値を用いてデータセットの被覆性を特徴付けることができる。深層ニューラルネットワーク（DNN）の内部ロジックはほとんどデータによってプログラムされているため、元データの統計的分布が非常に重要である。各特徴の被覆性は、DNN モデルの最終的な出力に大きな影響を与えるだけでなく、出力値がほとんど発生しないコーナーケースにも影響する。

例えば、ある属性 $x(n)$ が与えられたとき、与えられたテスト用データセット T が $[low_n, high_n]$ の範囲をどれだけ完全にカバーしているかを測定するのが k 区間被覆性（k-multisection coverage）である(nは n 番目の属性を指定し、$1 \le n \le m$で、mは T の属性の個数である)。この値は以下の通り定める。 $[low_n, high_n]$ の範囲を $k>0$ で k 等分する（したがって k 個の区間ができる)。また S_i^n により、$1 \leqq i \leqq k$ のとき、i 番目の区間内の値の集合を表す。$x(n) \in S_i^n$ ならば、i 番目の区間はこのテスト入力 x で被覆されている。そこで、あるテスト用データセット T と特徴 $x(n)$ に対して、その k 区間被覆性 は、T が被覆する区間数と区間の総数（我々の定義では k）の比と定義する。ある特徴量 n の k 区間被覆性を次のように定義する。

$$KMCov[x(n)、k] = \frac{|\{S_i^n \mid \exists x \in T: x(n) \in S_i^n\}|}{k} \quad (2)$$

さらにテスト集合 T の k 区間被覆性を次のように定義する。

$$KMCov[T、k] = \frac{\sum_{1 \le n \le m} |\{S_i^n \mid \exists x \in T : x(n) \in S_i^n\}|}{k * m} \quad (3)$$

ニューロンベースの被覆性

ニューロンベースの被覆性メトリクスとして基本的なものは Neuron Coverage (NC) [28]がある。これは元々テストデータの生成を自動化するために提案されたものである。これは、テスト用データセットの少なくとも 1 つの入力によって活性化されたニューロンの割合として定義される。

例えば、N個のニューロンからなる学習済み DL モデルを D とする。入力 x の D に対する Neuron Coverage は次式で与えられる。

$$NC(x) = \frac{|\{n \in N \mid activate(n, x)\}|}{|N|} \quad (4)$$

ここで、 activate(n, x) は、x を D に与えた時に n が活性化される場合に、かつその場合にのみ真となる。

その他のニューロンベースの被覆性メトリクスとしては、K-Multisection Neuron Coverage (KMNC), Neuron Boundary Coverage (NBC), Neuron Activation Coverage (NAC), Strong Neuron Activation Coverage (SNAC) などが文献 [29] に記載されている。

Surprise Coverage

Surprise Coverage (SC) [19]は、データの意外性に基づく新しい被覆性メトリクスの一種である。例えば、文献では、距離ベースの Surprise Adequacy（DSA）がデータの多様性と新規性を表現するために用いられ、コーナーケースデータを検出するのに有用であることが示されている。そこで、ここではバケッティングを用いて意外性の値空間を離散化し、距離ベースの Surprise Coverage (DSC)を定義する。ある上界 U と、 $(0、U)$ を分割した n 個の SA 区間からなるバケット $B = \{b1, b2, \ldots b_n\}$があるとき、入力の集合 X に対する SC は以下のように定義される。

$$SC(X) = \frac{\left|\left\{b_i \middle| \exists x \in X : SA(x) \in \left(U \cdot \frac{i-1}{n}、U \cdot \frac{i}{n}\right]\right\}\right|}{n} \quad (5)$$

SC の高い入力セットは、学習時に見たものと似たもの（＝SA が低い）から、学習時に見たものとは全く異なるもの（＝SA が高い）までを含む、多様な入力セットである。DL システムの入力セットの体系的な多様化を図る必要がある場合、SA は指標として有用である。、最近の結果 [30]では、より遠いテスト入力は例外を引き起こす可能性が高いが、テストにはあまり有効でない場合があることが示されている。

8.2.4 B-2: データセットの均一性

外観検査に限らず多くの用途では、正しい判断により確実に回避すべきリスクに対応する属性値の組合せについて十分な学習データを用意する必要があるだけでなく、「実世界の分布とは違っても敢えて重要なケースに十分な訓練データを用意するか、実世界の分布に忠実に沿った分布を持つ訓練用データセットで学習させるか」という判断が強く求められる。

数学的記述

上記の定義によれば、データの均一性を評価する簡単な方法は、データの既知の分布を仮定して実現することができる。実データの分布が $F_0(x)$ であり、与えられた学習データまたはテストデータが分布関数 $F_1(x)$ を持つと仮定すると、これら二つの分布関数の差は均一性を評価する指標となり得る。一般に、Kullback-Leibler divergence [31]は、この差を計算する良い方法であり、次のように定義される。

$$\boldsymbol{DKL(F_0 \| F_1) = \int_{x \in Cov} F_0(x) \ln \frac{F_0(x)}{F_1(x)} dx} \tag{6}$$

訓練用データセットとテスト用データセットの間の均一性

データの均一性は、データの分布が均一であることを意味するが、AI の品質に影響する

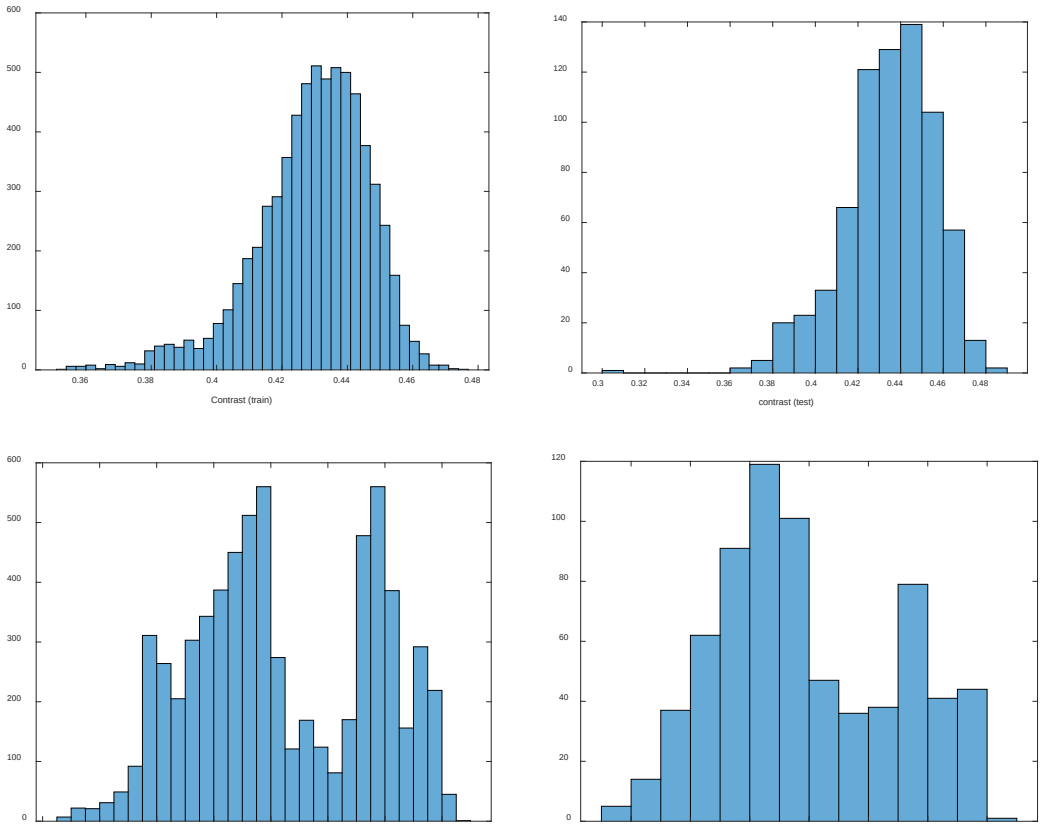


図 54 訓練用データセットとテスト用データセットにおける属性の分布

有効な評価手法の一つとしては、訓練データとテストデータの間の類似性もある。例えば、鋳造欠陥データに関して、画像の記述属性として最も一般的な画像の輝度とコントラストを例にとり、この 2 つの属性の学習データとテスト用データセットでの分布を求めると図 54 のようになる。

訓練時の属性分布とテストの際の属性分布はよく似ており、この結果、最終的な機械学習による外観検査システムも同様の性能を発揮する。

ケース間に渡る均一性

一方、データセットの均一性は、リスクの高いケースに対して十分なデータ量を保証すること、すなわち一般的なケースと同等のデータ量を保証することを求めている。これはつまり、異なるケース間での均一性を意味する。データが均一でない場合、AI の性能に大きな影響を及ぼす。例えば、データの不均衡の問題は、分類アプリケーションの精度を低下させる。一般に、実世界から収集したデータは不均一なため、小さなケースの間にもデータ被覆性にばらつきが生じる。外観検査の問題は主に分類に関わるため、データの不均一性や偏在の問題は AI の品質を保証するために重要である。

8.2.5 B-3: データの妥当性

この版のリファレンスガイドでは本品質は検討対象としていない。

8.2.6 C-1: 機械学習モデルの正確性

外観検査における正確性は、欠陥を正しく検出することと、欠陥の種類を正しく認識することを意味する。正確性に関する AI 品質評価には、8.2.1 節で挙げた、外観検査の KPI（適合率、再現率、正解率など）を用いることができる。

一般的な正確性メトリクス

外観検査に関わる主要な問題は、欠陥を異なる欠陥タイプに分類することである。そして、一般的な分類評価指標は、機械学習ベースの外観検査システムにも適用できる。

表 32 混同行列

		観測された状態	
		正例と観測	負例と観測
予測された状態	正例と予測	TP	FP
	負例と予測	FN	TN

さらに、混同行列に基づくメトリクス [32]は、分類やクラスタリングなど、イベントやラベルを出力形式とするモデルを対象とした分類評価にも利用することができる。混同行列は表 32 のように定義される。

上の表から、真陽性イベント（TP）、偽陽性イベント（FP）、偽陰性イベント（FN）、真

陰性イベント（TN）という 4 つのイベントが定義される。これらのイベントの数に応じて、再現率、適合率、正解率などの指標を定義することができる。以下にその定義を示す。

正解率（Accuracy）：正解率は、以下の式で与えられる。

$$\boldsymbol{Accuracy = \frac{TP + TN}{TP + TN + FP + FN}} \tag{7}$$

しかし、正解率には問題がある。二種類の間違いのどちらもコストは同じだと想定している。正解率が 99%であることは、問題によって、優秀、良い、平均的、悪い、ひどいのどれでもあり得る。

$$\boldsymbol{Recall = \frac{TP}{TP + FN}} \tag{8}$$

再現率（Recall）：再現率は、正しく分類された正例の総数を正例の総数で割った比率として定義される。高い再現率は、クラスが正しく認識されている（FN が少ない）ことを示す。

$$\boldsymbol{Precision = \frac{TP}{TP + FP}} \tag{9}$$

適合率（Precision）：適合率の値を得るには、正しく分類された正例の総数を予測された正例の総数で割る。高い適合率は、正例と分類された例が本当に正例である（FP の数が少ない）ことを示す。

- **再現率が高く、適合率が低い**：これは、ほとんどの正例が正しく認識されている（FN が低い）が、偽の正例が多いことを意味する。
- **再現性が低く、適合率が高い**：これは、多くの正例を見逃したが（FN が高い）、正例と予測したものは本当に正例であることを示している（FP が低い）。

2 つの指標（適合率と再現率）を併用する代わりに、その両方を表す指標があると便利である。F スコアの計算には、適合率や再現率が極端な値の時により悪い値となるように、算術平均ではなく調和平均を使用する。F スコアは、常に適合率や再現率の小さい方の値に近くなる。

$$\boldsymbol{F_measure = \frac{2 * Recall * Precision}{Recall + Precision}} \tag{10}$$

鋳造欠陥のデータを例にして、外観検査に関する様々なモデルの正しさを分析することができる。第一の品質、**問題領域分析の十分性、**で述べたように、鋳造欠陥データに基づく外観検査は、標準的な 2 値分類問題である。ここでは、3 つの分類モデル、CNN、VGG16 [33]と ResNet34 を適用し、外観検査システムを構築する。その訓練パラメータを以下の表に示す。

表 33 鋳造欠陥データに対する 3 モデルのパラメータ

	バッチサイズ	エポック数	学習率	訓練精度(%)

CNN	64	10	0.0002	99.17
ResNet34	64	10	0.0002	99.44
VGG16	64	10	0.0002	92.76

この表では、各 AI モデルの正確性を分類精度で表現し、訓練精度のみを示している。次に、ガイドラインに従って、これらの訓練済みモデルをテストデータで評価し、正確性の品質に関して性能を調査する。正確性の分析結果を以下に示す。

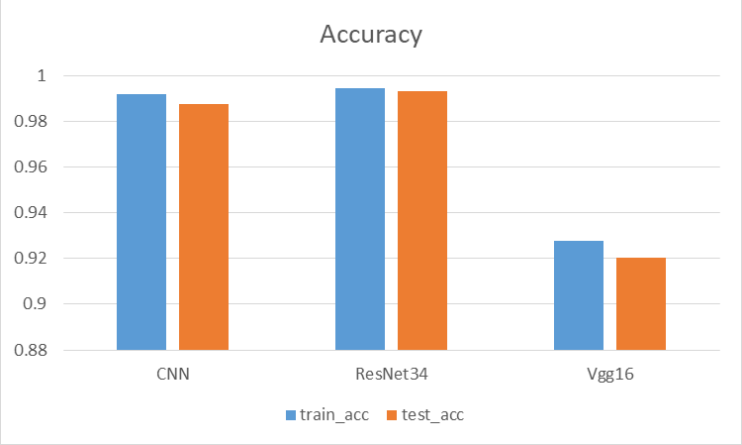


図 55 訓練工程とテスト工程における 3 モデルの精度

コーナーケースデータ検出における正確性

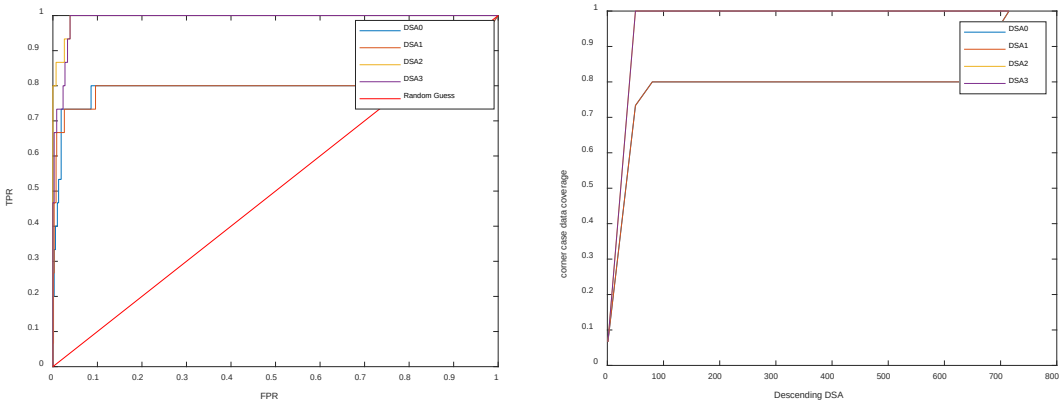
機械学習モデルの正確性の定義によれば、この品質には正しい動作に対する評価だけでなく、誤った動作に対する評価も含まれている。そこで、ここでは、**問題領域分析の十分性**(8.2.1 節)で述べた、コーナーケースデータという、誤りを誘発するデータの評価に主眼を置いた新たな評価指標を提案する。コーナーケースデータ検出に関する AI モデルの正確性を定量的に評価するため、ここでは、コーナーケースデータの被覆性として、以下のような指標を提案する。

$$\boldsymbol{cov}(\boldsymbol{v_{th}}) = \frac{\boldsymbol{card}(\{\boldsymbol{d}|\boldsymbol{d} \in \boldsymbol{CD}, \boldsymbol{DSA}(\boldsymbol{d}) > \boldsymbol{v_{th}}\})}{\boldsymbol{card}(\boldsymbol{CD})} \times \mathbf{100\%} \qquad (11)$$

ここで、 v_{th} は与えられた閾値、*CD* はコーナーケースデータのデータセット、*card()* はカーディナリティ、コーナーケース検出には DSA を用い、{d}は与えられた閾値より大きな DSA 値を持つ、検出した全てのコーナーケースデータからなる集合を表す。

❖ 例

鋳造欠陥のデータを対象に、元データの中からコーナーケースを検出するために、いくつかのコーナーケース記述子を適用することができる。ここでは距離ベースの Surprise Adequacy（DSA）を使う。そして、2 つの畳み込み層と 2 つのフルコネクション層からなる一般的な CNN モデルを構築し、外観検査を行う。モデル化と訓練の結果、最終的な検査精度はこの鋳造データで 97.90%に達する。DSA に基づくコーナーケースデータ検出のパフォーマンスを以下の図に示す。



(a) ROC 曲線　　(b) コーナー事例データ

図 56 コーナー事例データ検出の性能

図 56 では、コーナーケース検出の性能分析のために 4 種類の DSA [27]を用いている。図 56 (a)では、コーナーケース検出の性能を評価するために、FPR（偽陽性率）と TPR（真陽性率、すなわち再現率）の値によって ROC 曲線をプロットしている。また図 56 (b)は、提案したコーナーケースデータの被覆性を計算したもので、X 軸は DSA の降順でデータをソートしたものである。X 軸上の特定の値 Xt に対応するグラフ上の点は、DSA 値が最も大きい点 Xt 個を考慮して行ったコーナーケースデータの被覆性の計算、すなわち外観検査における誤動作の割合の計算の結果を示している。

8.2.7 C-2: 機械学習モデルの安定性

機械学習を用いた外観検査システムの安定性には、以下のような問題が起こりうる。まず、敵対的な攻撃は、外観検査システムの安定性に影響を与える。例えば、訓練済み外観検査システムに入力される欠陥画像に少量のノイズを加えると、システムの挙動が大きく変化し、すなわち安定性が破壊されることがある。このようなノイズには、カメラレンズの汚れなど自然界のランダムなノイズと、悪意ある攻撃による敵対的な摂動がある。第二に、AI モデルの頑健性は、外観検査システムの安定性を左右する要因でもある。例えば、機械学習ベースのモデルが過適合している場合、そのモデルを用いた外観検査システムは過敏な動作をする。すなわち、学習データの分布から外れたデータに対して誤判断しやすくなる。

数学的定義

ソフトウェア工学の用語では、頑健性の標準的な表記は、以下の通り [34]である。「システムまたはコンポーネントが、無効な入力やストレスの多い環境条件のもとでも正しく機能する度合い」これは、上記の安定性の説明と類似している。この定義を数学的な言語に変換すると、以下のようになる。

定義 1（頑健性） [35]　S を機械学習システムとする。S の正しい出力を E(S)とする。データ、学習プログラム、フレームワークなどの機械学習の構成要素に摂動を与えた機械学習シ

ステムを δ(S)とする。機械学習システムの頑健性は、E(S)と E(δ(S))の差に現れる。

$$r = E(S) - E(\delta(S)) \tag{12}$$

ここで、r は頑健性に関わる測定値の一つで、摂動の存在下での ML システムの正しい出力の回復力を見るものである。

頑健性のサブカテゴリーとしてよく知られているのが、敵対的頑健性というものである。敵対的頑健性は、検出されにくいように設計された摂動を扱う。ここでは、局所的な敵対的頑健性を紹介する。

定義 2（局所的敵対的頑健性）[35] x を ML モデル h のテスト入力とし、x に敵対的摂動を与えて生成した別のテスト入力を x' とする。モデル h は、以下を満たす時、入力 x に対して δ 局所的敵対的頑健である。

$$\forall x' : \left|\left|x - x'\right|\right|_p = \delta \rightarrow h(x) = h(x') \tag{13}$$

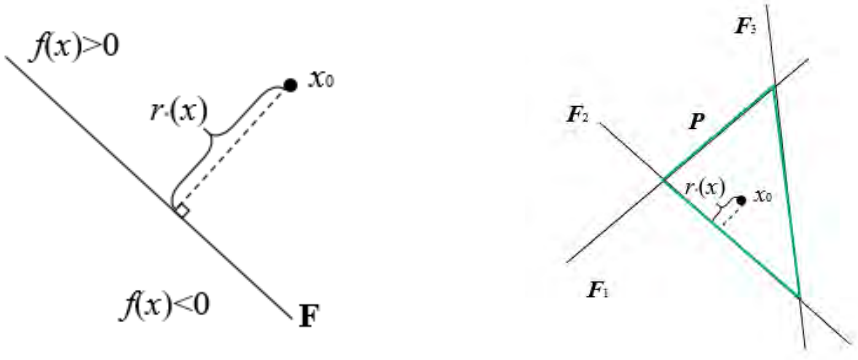
ここで、$\left|\left|\cdot\right|\right|_p$ は距離測定の p-ノルムを表す。

頑健性測定方法 [36]

頑健性の定義によると、頑健性の測定は、実際には、テストデータが異なる判断をする原因となる最小の摂動を決定することである。以下の 図 57 (a) に示した線形 2 値分類器の場合、与えられたテスト点 x_0 に対する DL モデルの判断が変わる最小摂動は、x_0 から超平面 F までの最小距離であり、以下の形式で記述される。

$$\begin{aligned} &\mathbf{r}_*(\mathbf{x_0}) = \mathbf{argmin}\, \|r\|_2 \\ &s.t. sign(f(x_0 + r)) \neq sign(f(x_0)) \end{aligned} \tag{14}$$

同様に、2 値分類器の集合で構成されるマルチクラス分類器でも、図 57 (b) に示すように、x_0 に対する DL モデルの頑健性測定を、x_0 から分類境界までの最小距離として算出することが可能である。



(a) 線形 2 値分類器　　(b) 複数の 2 値分類器からなる多クラス分類器。

図 57 分類器の頑健性

さらに、線形でない任意の分類器に対しては、反復処理によって頑健性の測定値を算出

することができる。これは、反復の各段階において、分類器の微分部分を線形とみなすことができるからである。

コーナーケースデータを考慮した頑健性測定

さらに、安定性品質が過適合を回避する必要があることを考慮すると、コーナーケースデータが AI モデルの頑健性分析に与える影響 [37]を考慮することも有用な応用例であると考えられる。ここでは、正確性分析とは異なり、以下に示すように、訓練データからコーナーケースデータを検出し、検出されたコーナーケースデータを削除してモデルの再学習を行うことを考える。

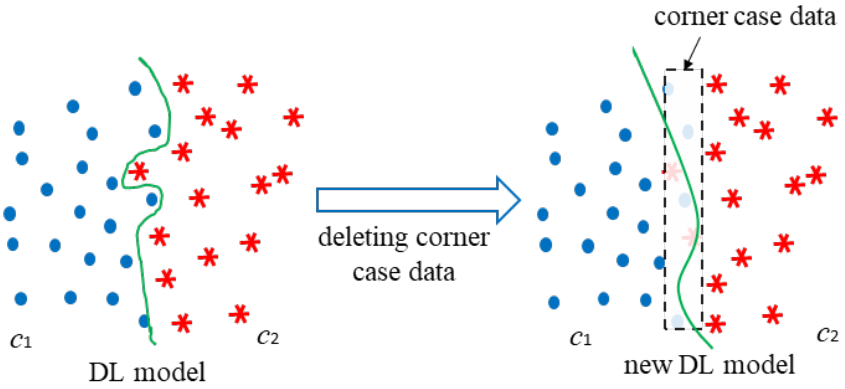


図 58 頑健性の向上

こうすることで、訓練済みモデルはより頑健なものになる。境界点（コーナーケースデータ）の影響が排除され、モデルの過適合のリスクが低減するためである。鋳造欠陥データを例として、この外観検査問題に対する一般的な CNN モデルを構築し、上記で述べた手法を適用して実験を行い、このモデルの頑健性を測定した。頑健性の測定結果は以下の表の通りである。

表 34 鋳造データに対する頑健性測定

		L1	L2	L∞
モデル 1	最小値	0.0010	0.0088	0.3115
	最大値	0.7421	7.0115	264.2126
	平均値	0.1890	1.9058	69.4177
モデル 2	最小値	0.0024	0.0240	0.8842
	最大値	0.7653	7.0000	261.8947
	平均値	0.2026	1.9730	72.4405

AI モデルの頑健性を向上させるために、ここで前述の手法を適用した。例として、元の学習データから上位 k 個のコーナーケースのデータを削除し、残りのデータで外観検査モデルを再訓練した。そして、再訓練したモデル（モデル 2）と元のモデル（モデル 1）の頑

健性を比較すると、表 34 のようになった。頑健性の測定値の計算には 3 種類のノルム L1、L2、L∞を用いた。どのノルムを用いた場合にも、再訓練後のモデルの頑健性の測定値の方が大きくなっており、コーナーケースのデータを取り除いた学習セットに基づき AI モデルを再訓練することで、モデルの頑健性を向上させることができるという結論を示している。

8.2.8 D-1: プログラムの信頼性

この品質には、モデル、データ、実行環境の信頼性を保証する必要がある。例えば、鋳造欠陥データのような、いくつかのデータセットについては、Web サイトからオープンソースのコードが公開されていることがある。それらを最終的な外観検査システムの開発に直接利用するのであれば、開発者の責任で十分な品質を確保する必要がある。

信頼性評価の方法として、従来のソフトウェア工学の品質管理手法を機械学習による外観検査システムにも適用することが可能である。ただし、この品質保証においては、以下の点を考慮する必要がある。

まず、開発環境と実運用環境との整合性である。外観検査問題のプログラムを開発する際、環境として利用できる選択肢は多数ある。最終的なソフトウェアが正常に実行できることを保証するためには、OS、エンジン、バージョンなどの環境を決定する必要がある。例えば、機械学習による外観検査システムを含め、一般的な深層学習システムは、Python をベースに開発することが可能である。

第二に、外観検査用のハードウェアは、システムの信頼性を左右する重要な要素である。例えば、外観検査のアプリケーションでは、入力は一連の画像であるため、学習処理には通常、計算機の GPU、さらにはサーバーやクラウドコンピューティングが利用される。しかし、このシステムを欠陥検出の実運用に移行する場合、そこで用いるプラットフォームがこれらのハードウェア要件を同様に満たす保証はなく、満たさなければ、外観検査の動作は実現できない。従って、信頼性保証のためには、あらかじめハードウェア要件を考慮しておく必要がある。

第三に、メモリ要件も信頼性品質に影響を与える可能性がある。この要因は、入力画像サイズ、パラメータのサイズ、訓練やテストのバッチサイズを決定する可能性がある。もし、訓練や開発には大量のメモリを使用するが、テストには少量のメモリしか使用しないのであれば、外観検査システムのディペンダビリティは保証できない。

8.2.9 E-1: 運用時品質の維持性

運用時品質の維持性とは、運用開始時に満たされていた内部品質が運用期間中も維持されることを意味する。この内部品質が実現されているなら 1) 外部環境の変化に十分に対応でき、 また、2) その対応のために訓練済みの機械学習モデルに加えた変化のせいで品質が劣化することを防ぐことができる。

この新しい AI 品質に関する定義に基づいて、機械学習ベースの外観検査システムでは、2 つのテーマを検討する余地がある。

まず 1 つ目は、追加学習をどのように行うかである。例えば、外観検査において、新種の欠陥が見つかった場合、機械学習ベースのシステムは、その種類の欠陥に適応するように再訓練する必要がある。追加学習によって、外観検査システムは新たな状況に対応できるようになる。

2 つ目は、反復学習をどのように行うかである。この要件は主に適応学習やオンライン学習で適用される。与えられた欠陥データセットに基づく学習済みヴィジョン検査モデルは、データ量が常に有限であるため、必然的に不完全なものになる。そこで、システム性能を向上させるため、新しい欠陥画像やコーナーケースデータをシステムに入力するが、その際、パラメータを若干調整することができる。

9 郵便番号の分析

9.1 はじめに

ここでは、郵便番号の分析の問題について説明する。この問題は、書かれた数字を識別するもので、よく知られている MNIST の数字分類問題に非常によく似ている。以下、この問題の詳細と可能な解決策を説明する。

本節は、郵便番号分析の解決策を見出すために検討したアイデアや分析の記録である。ここでは、産総研の MLQM ガイドラインに基づく検討全体を説明する。本節の目的は、単に郵便番号の問題を分析するだけでなく、ガイドラインの妥当性を検証することにある。本節で報告する事項は、以下の通りである。

- 郵便番号分析問題の詳細と実例分析
- データマネジメントの課題と方法
- 問題への AI の適用
- データや AI モデルの検証に関する手法や考え方
- 今後のメンテナンスと取組みについての案

9.2 ビジネス要件

9.2.1 問題定義（ユースケース）

AI モデルは、封筒の郵便番号を機械で読めるようにして、番号に基づいて手紙や小包を仕分けられるようにするために、さまざまな手書き文字に対応した数字を（1 つずつ）識別する。郵便番号の各桁を独立に識別するため、AI の傍らで支援ソフトウェアが動作して、一桁ごとの画像を取り出して AI の入力とし、また AI の出力をまとめる。ここでは、本製品の利用者として、郵便局（郵便事業者）を想定する。

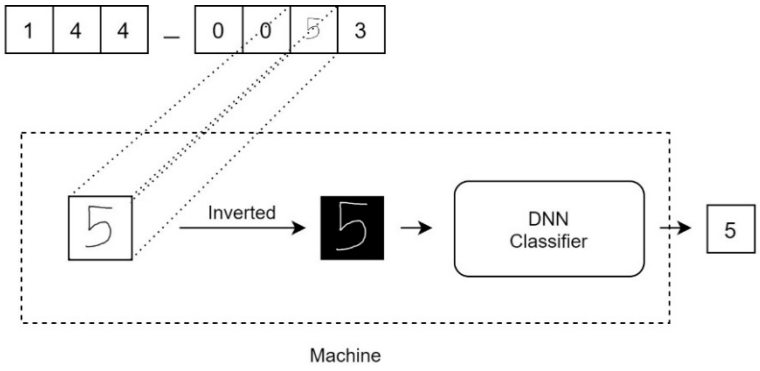


図 59 郵便番号解析機のワークフローの概要

図 59 に示すように、まず、コードの各番号を順次撮影し、前処理と色反転を行う。前処理には、数字を中央に配置する、コントラストを強くする、などが含まれる。その後、学習させた分類器に画像を渡し、予測結果を出力する。以上が郵便番号検出装置の簡単な説明である。以下では、機械の各部にどのような仕様が必要なのかを説明する。

9.2.2 背景

郵便番号は、郵便事業において住所を特定するのに役立つ簡略符号である。多くの国では、郵便番号は都市名と州・地域を特定するのに十分な情報を含んでおり、詳細な住所表記の必要性を最小限に抑えるだけでなく、郵便物の自動分類を短時間で行うことができる可能性を開く。これは、郵便事業会社や政府機関にとって、多くの時間と労力の節約につながる。

9.2.3 目的・目標

本節では、主に手書き郵便番号検出の AI 実装を中心に報告する。しかし、モデルの入力を集めて準備する入力システムと、モデルの出力を有効に利用する出力システムが存在し、画像の前処理や予測の後処理も開発段階で定義し、開発することになる。しかし、AI モデルだけに焦点を当てると、その目的は次のように定義できる。

- 学習させた AI モデルが、画像から手書き数字の数字を識別する。
- このモデルでは、入力画像に含まれる様々な外乱（開発者が選択し、ステークホルダーが承認した範囲のもの）を考慮する。
- AI モデルは高速に動作するよう、あまり複雑でないことが必要である。

9.2.4 この製品のステークホルダー

手書き郵便番号自動検出システムには、以下のステークホルダーがいる。

- 郵政事業者（出資者や監督官庁を含む、システムの利用者）
- 従業員（システムの操作者）
- ユーザー/顧客（郵便サービスの利用者）

9.2.5 ステークホルダーの初期要求

- すべてのステークホルダーは、正確な出力が得られることを望んでいる。
- 郵便事業者は、経済市場で効率的に競争するために、高速な推論時間を求める。
- 郵便事業者は、運用コスト削減のために、高いコストパフォーマンスも求める。
- サービス提供者（従業員）の立場からすると、使い方が簡単でわかりやすいことが

必要である。

- また、顧客にとって使いやすいサービスにするためには、AI は、多様な画像に対応して、入力の柔軟性を高める必要がある。例えば、異なるインクで書かれた文字や、異なる色の紙の背景に対応するなど。
- 郵便事業者は、市場のできるだけ多くの顧客からの郵便物や小包の処理効率を上げるために、多種多様な手書きの受け入れを求めるかもしれない。

9.2.6 ビジネス要件の詳細

機能要件

- AI が手書き数字（0〜9）を検出する。
- 固定サイズのグレースケール/RGB 画像を入力として受け取る。
- もし、AI が数字認識結果に確信がない場合は、ログを残し、後で人が確認できるようにする。

非機能要件

- 高速な処理。郵便事業者は、認識システムが 1 秒以内に出力することを求めている。
- ノイズに対処でき、様々なスタイルの手書き文字に対応可能なモデルであること。

前提条件

- 手書き数字認識の際、AI モデルは、できるだけノイズのない明瞭な画像を受け取ること。

依存事項

- 保守の際、システムが確信を持てない事例や障害が発生した事例を監視する。これは、運用段階でのシステム改善に役立つ。

制約事項

- データの制約。今回の初期の製品開発では、開発者は一般に公開されているデータセットしか使えない。

考慮しない事項

- 活字フォントは、この AI モデルの対象外とする。

リスクと安全に関する懸念

- 郵便番号の誤認識は、郵便物の誤配送を引き起こし、金銭的・時間的な損失につながる。

9.2.7 外部品質に関する要求事項

上記のビジネス要件に基づき、達成すべき外部品質をいくつか設定する。MLQM ガイドラインに沿って設計することで、開発者や評価者にとってより分かりやすい外部品質と

なる。

安全性

- 本製品による人体へのリスク・事故の懸念はない。
- この製品による経済的損失に関する安全性の懸念がある。これはモデルの性能と密接に関係している。

パフォーマンス

- このモデルは、定義された KPI について、運用フェーズでも開発フェーズと同等のパフォーマンスを示す必要がある。
- クラスごとの性能は総合性能と同等であることが望ましい。

公平性

公平性に問題はない。

9.2.8 外部品質特性レベルを定義する

表 35 実現すべき外部品質特性レベル

外部品質	補足説明	想定される深刻度	実現すべきレベル
安全性	人的リスクに対する AI 安全性レベル	人体への物理的な被害はないと思われる。	AISL 0
	経済的リスクに対する AI 安全性レベル	資産への重大な損害。人による監視で回避可能。	AISL 1
パフォーマンス	AI 性能レベル（総合的）	製品やサービスがシステム運用のため一定の KPI を満たすこと。	AIPL 2
	AI 性能レベル（クラス別）	製品やサービスがシステム運用のため一定の KPI を満たすこと。	AIPL 2
公平性	AI 公平性レベル（総合的）	製品にもサービスにも公平性に関する要件はない。	AIFL 0

9.2.9 おわりに

ビジネス要件に関する本節では、ビジネスオーナーの視点を示した。ここに示したビジネス上の要求と製品の外部品質は、架空の想定である。実際の現場では、ビジネス担当者やマーケティング担当者と製品開発者がチームを組んですべての意思決定を行い、各外観品質について実現すべきレベルを定義することになる。本節が、開発者とビジネスオーナーと

のコミュニケーションと、製品要件の決定に役立てば幸いである。

9.3 製品仕様

このパートでは、最終製品の仕様や AI ソリューションに対するクライアントの期待について説明する。郵便番号の分析という特定の問題に対しては、以下のような仕様や記述が可能である。

9.3.1 データ関連仕様

- **画像データによる分類**　画像データのみで、数字の識別を行う必要がある。スキャンした文書からボックスを切り取ることで画像を得ることができる。これらの画像は、計算を容易にするためにグレースケール化したり、反転させたりすることができる。
- **文字を書くときに使うインク**　一般に何か書くときには鉛筆やペンが使われるが、鉛筆は黒鉛の等級が色々あり、ゲルのペンは文書上にインクがにじむ場合がある。そこで、数字を書く手段は、黒いボールペンとする。
- **入力（手書き）の禁止パターンを宣言する**　手書き入力の様式は無数に存在するため、曖昧で識別困難なため受け付けない入力パターンを定義しておく必要がある。例えば、'9'のループは必ず閉じないと'4'と同じようなパターンになる。

9.3.2 モデル仕様

- 学習の種類：教師あり学習
- AI モデルの種類：分類
- モデルのアーキテクチャ：CNN
- **実行するタスク（数字の識別）**：郵便番号は数字のみで構成されている。そのため、この製品は 0 から 9 までの 10 種類の数字を識別できる必要がある。

9.3.3 KPI 仕様

- **精度**：精度、再現率、適合率、F スコアなど
- **安定性/頑健性：**変動に対する頑健性、距離による surprise adequacy（DSA）など。

9.4 データセットの紹介

9.4.1 データセットの探索

郵便番号の検出や数字認識は、AI による教師あり分類問題である。そのため、学習とテストのためになんらかのデータセットを用いることになる。データセットに関しては、以下のいずれかの対応が考えられる。

オープンソースのデータセットであれば、なんでも利用できる。例を以下に示す。

- The MNIST database [38]
- USPS dataset – Handwritten digits [39]
- ARDIS - スウェーデンの歴史的手書き数字データセット [40]

複数のデータセットを組合せて使ってもよい。また、必要に応じて独自のデータセットを構築することもできる。

このうち、MNIST は手書き数字のデータセットとして最もよく知られているため、本節ではこれを用いて分析を行う。

9.4.2 MNIST データセット

MNIST データセットには 7 万枚の画像データがあり、6 万枚が訓練データ、残りがテストデータである。これらは、28×28 の大きさのグレースケール画像である。データセットの説明によれば、250 人の様々な書き手が作成に関わった。また、ピクセルの重心を計算して、数字が画像中央に来るよう寄せてある。

データセットを選択し、モデルを訓練する際には一般に、ソリューションの対象地域を決める必要がある。郵便番号検出の問題でも、世界のどの国、どの地域で使うソリューションなのかを明らかにする必要がある。それがわかれば、その地域に基づいたデータセットを構築するべきである。MLQM ガイドラインが示す評価手法は世界共通だが、データセットはそうではない。MNIST は米国で作成された手書き文字データセットなので [41]、MNIST で訓練した分類器は、米国人向けになる。

9.4.3 入力データのサンプル

モデルへの入力データ（画像）の一例を図 60 に示す。28X28 のグレースケール画像で、色が反転しており、コントラストが強い。これは MNIST データセットのテストセットからの画像の 1 つである。この画像は MNIST データセットに含まれるテスト画像の一つであり、数字は明瞭であるが、使い方を考えると、うまく中央に寄せてあるとはいえない。そういった、様々な理由から、画像の分布や向きを分析して後の利用に備える必要がある。この

部分を MLQM 評価と呼ぶ。

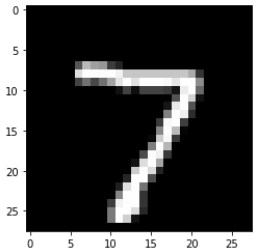


図 60 MNIST データセットからの郵便番号解析器への入力例

9.5 MLQM ガイドラインを用いた品質保証手順

ここでは、郵便番号検出に関するすべての分析結果を紹介する。品質評価項目は 9 つある。

- 問題領域分析の十分性
- データ設計の十分性
- データセットの被覆性
- データセットの均一性
- データの妥当性
- 機械学習モデルの正確性
- 機械学習モデルの安定性
- プログラムの信頼性
- 運用時品質の維持性

9.5.1 A-1：問題領域分析の十分性

定義

問題領域分析の十分性とは、機械学習モデルへの入力となる実データのあらゆる特性を分析することである。実世界で起こりうるすべてのケースを網羅することが第一の目標である。また、不可能なケースがあれば、そのシナリオも分析・定義する。

ソリューションに着手する前に、まず要件を決める必要がある。郵便番号の検出では、画像から手書きの数字を識別することが大きな目標であるが、それだけではない。認識するべき数字は 10 種類しかないが、手書きの数字となると、その種類は膨大なものになる可能性がある。そこでまず、入力画像にありそうな特徴を定義し、問題領域を設定する必要がある。郵便番号の検出については、AI を使ったソリューションになるので、機械が人間の能力と同等であることを期待することになる。言い換えると、「人間の目が識別できる数字はすべて、機械も識別する必要がある」。次に、各要件を順に説明する。

あり得るすべてのクラスのデータ

データセットでは、訓練セットとテストセットの両方で、すべての数字（0、1、2、3、4、5、6、7、8、9）の均一な分布が必要である。

適切な特徴を属性として選ぶ

同じ数字の画像や手書き文字に違いが生じる理由は色々ある。ここでは、そのすべてを列挙しようと試みた。

I. **数字の位置**　画像の枠の中での数字の位置のことである。



図 61 数字の位置の違い

II. **面積**　画像に対する数字の大きさを意味する。

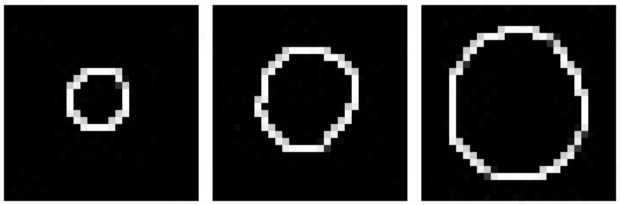


図 62 数字の面積の違い

III. **長さ**　筆跡の長さのことである。下図では、'1'は短く、'8'は長い。

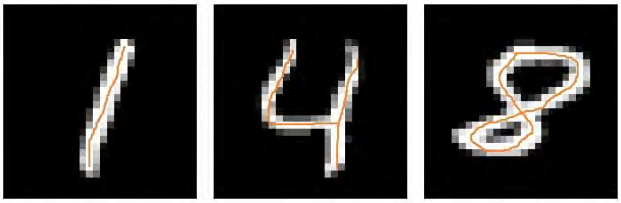


図 63 数字の長さの違い

IV. **明度／コントラスト**　画像の明るさや鮮明さを意味する。



図 64 数字のコントラストの違い

V. **傾き**　数字の向きを指す。



図 65 数字の傾きの違い

VI. **太さ**　見ての通り、筆跡の太さのことである。

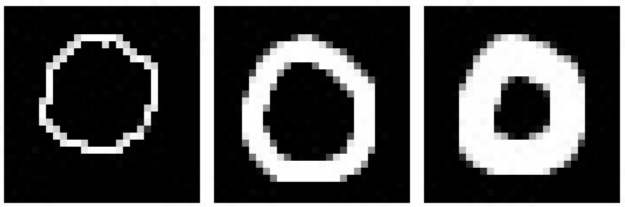


図 66 数字の太さの違い

VII. **手書きのスタイル**　母集団の大きさに比例して、多様な手書き文字がある。この特徴は数字によって異なる。そのため、完全に分析しようとすると、複数(10)通りの異なる問題領域を作る必要がある。ここでは、数字「9」の異なる書き方の例をいくつか示す。次のステップでは、これらの書き方を記述するためのいくつかの特徴量とその値を定義する。



図 67 '9'の書き方の違い

対象とする値域の選択

次に、選択した各特徴量に対して、取り扱う値または値の範囲を定義する必要がある。ここでは、任意の手書き数字（グレースケール画像）に当てはまり、数字に依存しない特徴量 5 つと、数字に特化した問題領域 1 つについて説明する。

I. **数字の位置**　画像フレームは、図 68 のような 4 つの基本的な領域に分けることができる。この機能は、数字を画像の特定の領域から切り出せるなら、省略できる。

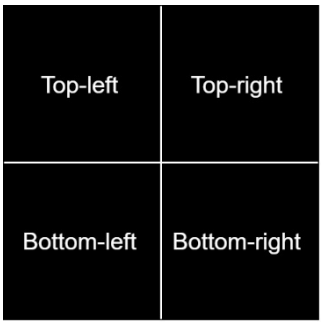


図 68 画像の 4 つの領域

II. **数字の大きさ**　画像に対するバウンディングボックスの比率は 76-100%にはならないと想定すると、小（画像枠に対して 0-25%）、中（26-50%）、大（51-75%）の 3 通りの数字の大きさを考えることができる。この機能は、数字を切りだしたり、余分なピクセルを周辺に足すなどして、数字の大きさを調整できれば省略できる。



図 69 数字の大きさの違い

III. **明度／コントラスト**　白や黒の明確さを表す指標である。コントラストが低いと、数字のエッジ検出が難しくなる。コントラストの種類は 2 つだけにしておく。

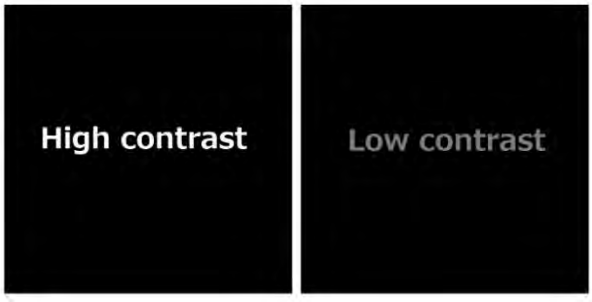


図 70 数字のコントラストの違い

IV. **傾き**　この特徴量は、まっすぐ、右に傾いている、左に傾いている、に大別できる。この特徴を角度で表現することも可能であるが、複雑になるし、詳細の扱いは学習過程に任せることができる。ここでは、向きの違い 3 つだけを考える。

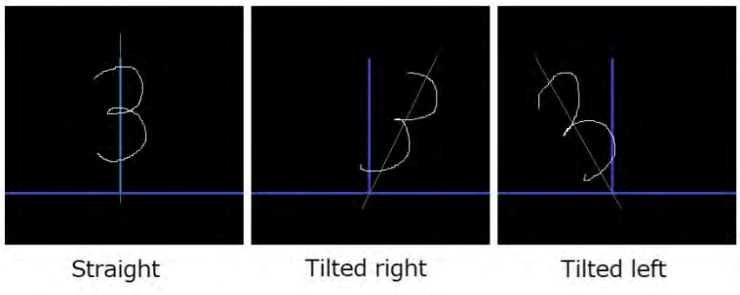


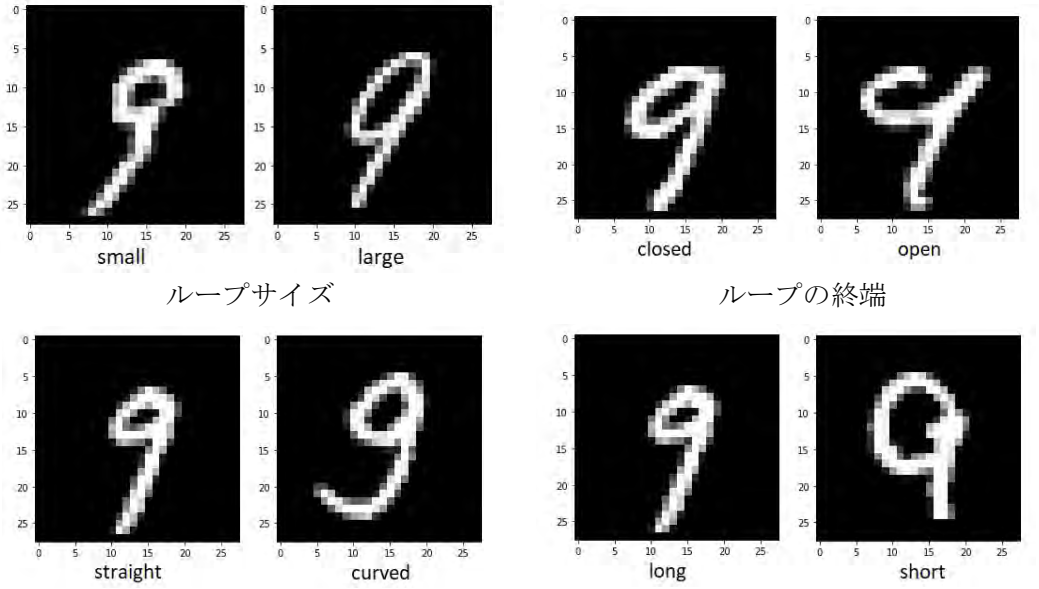
図 71 数字の傾きの違い

V. **太さ**　この特徴は、ペンのインク幅にのみ依存する。ここでは、線の太さを細い、普通、太い、の 3 段階に分けてみる。また、太さの近似値として非ゼロのピクセル数を考えれば、数値の値域を選択することも可能である。ここでは、想定する、ないし許容する数字ピクセルの比率を 5～20%程度とした。



図 72 数字の太さの違い

VI. **手書きのスタイル**　これは単一の特徴ではなく、どの数字固有の問題領域にも共通して存在する特徴である。特徴量の個数や値域が異なる 10 個の領域が存在する。ここでは、数字「9」に関して問題領域を詳しく示す。様々なタイプの手書き文字に基づいて、「9」を書くときに発生し得る特徴としては、以下のものが考えられる。



ループサイズ　　ループの終端

末尾の形状　　末尾の長さ

図 73 手書きのスタイルの違い

問題領域のまとめ

ここでは、問題領域全体を一目で分かるようまとめる。以上の分析の結果、問題領域は以下の通りとなる。

表 36 問題領域のまとめ

特徴量	特徴量の値

数字の位置	左上、右上、左下、右下（4 種）
数字の大きさ	小、中、大（3 種）
明度/コントラスト	高い、低い（2 種）
傾き	まっすぐ、右に傾いている、左に傾いている（3 種）
太さ	細い、普通、太い（3 種）
手書きのスタイル（数字依存）	'9'： ループサイズ：小、大（2 種） ループの終端：閉じている、開いている（2 種） 末尾の形状：直線、曲線（2 種） 末尾の長さ：長い、短い（2 種）

おわりに

品質保証における問題領域の分析はここで終了する。あらゆるあり得る実世界のデータについて、特徴量を割り振ることができる。次節では、属性や特徴の組合せのケースに注目し、その組合せの妥当性を確認する。

9.5.2 A-2: データ設計の十分性

定義

問題のケース、すなわち特徴の組合せは、さらにデータ分析を行う前に調べておく必要がある。それにより、後の分析で各ケースに対応するデータの有無が分かり、レアケースやコーナーケースを見つけるのに役立つからである。しかし、データの存在がまず確認できそうにない、不可能なケースもあり得る。ここでは、ここで扱う問題における、そのようなケースを取り上げ、データ分析の対象として有効なケースを選択する。

不健全な（絶対に発生しない）ケースの洗い出し

有効/健全なケースを選択するには、まず不健全なケースを特定する方が易しい。不健全なケースは、そこに関わる特徴が互いに依存しあっている場合に現れる。例えば

- 雨が降っていれば、地面が濡れている（車両運転データセット）。
- プールがない家では、プールの質は問われない（住宅価格データセット）。

郵便番号の検出では、問題領域に対して定義した特徴量は互いに独立である。したがって、すべての特徴の組合せが有効である。上記で説明した特徴空間に基づいて、組合せの数を算出できる。数字に依存しない特徴のみを考慮すると次のようになる。

- $Number\ of\ combinations = 4 \times 3 \times 2 \times 3 \times 3 = 216$

おわりに

問題ケース分析の結果、この報告では分析しきれないほど多くのケースが得られた。以下、そのうちの 1、2 ケースを分析する。

この問題で考えられる不健全なケースは一つだけある。データセット中の数字の注釈が

誤っていることである。

9.5.3 B-1: データセットの被覆性

定義

データセットの被覆性、すなわちデータカバレッジとは、特徴空間全体にわたってデータポイントが存在することを意味する。被覆性を高めること、あるいは被覆性基準を満たすことは開発者の仕事である。評価者としては、問題領域内の抜けている箇所を特定する必要がある。本節では、問題領域に対する被覆性を分析し、どんなデータが足りないかを示す。また、データカバレッジに関連する様々な措置もここで説明する。

データカバレッジの計測

ここでは、MNIST データセットを用いて、特徴量「**面積**」のカバレッジ分析を行う。この特徴は、数字を覆う最小の平行四辺形の面積として定義する。MNIST の画像データのサイズは 28X28 であるため、この特徴量の数学的定義域は[0、784]である。先に定義した問題領域では、面積の特徴量は 3 段階に分かれている。

- 小：数字は画像枠の 0-75 ピクセルを占める
- 中：数字は画像枠の 75-200 ピクセルを占める
- 大：数字は画像枠の 200-500 ピクセルを占める

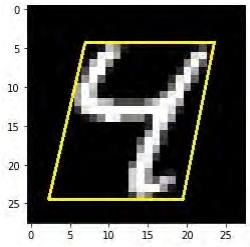
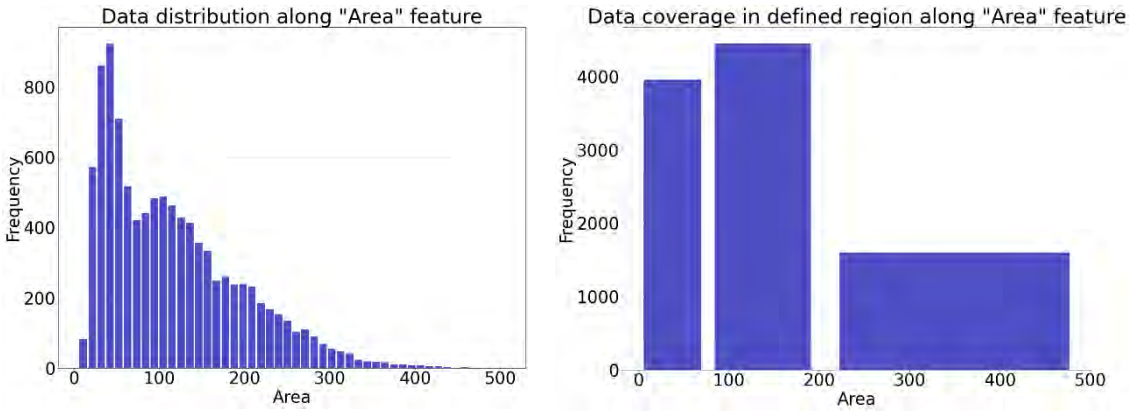


図 74 特徴量「面積」の指標を定義するバウンディングボックスの例

ここで、すべてのテスト画像について**面積**を計算したところ、以下の分布が得られた。



(a) **面積**のデータ分布　　(b) **面積**の分類ごとのデータカバレッジ

図 75 特徴量「面積」の分布

均等な分布ではないが、特徴空間に定義したすべての領域にデータが存在することがわかる。データカバレッジを定量的に示すと、以下の通りである。

- 小：3954
- 中：4447
- 大：1598

つまり、対象とする特徴空間では、すべての領域にデータがある。このカバレッジ計算は、8.2.3 節で定義されている K 区間被覆性のようなものである。

また、データセットで見つかった面積の範囲を取り出して、その特徴量の定義域と比較することもできる。これにより、データカバレッジに数値的なランクを与えることができる。数学的に表現すると、 $x \in dataset$ とすると、n番目の特徴について以下のようになる。

$$\boldsymbol{Cov}[\boldsymbol{x(n)}] = \frac{|\boldsymbol{max}\{\boldsymbol{x(n)}\} - \boldsymbol{min}\{\boldsymbol{x(n)}\}|}{|\,\boldsymbol{high_n} - \boldsymbol{low_n}|} \tag{15}$$

$$\boldsymbol{Cov}[\boldsymbol{MNIST}('\boldsymbol{size}')] = \frac{|\mathbf{500} - \mathbf{5.75}|}{|\,\mathbf{500} - \mathbf{0}|} = \mathbf{0.9885} \tag{16}$$

テスト用データセットにおける特徴量**面積**のカバレッジを計算すると上記のようになる。このカバレッジ分析は、従来のカバレッジと呼ぶことにする。**郵便番号分析**の先行研究では、データのカバレッジを計算するために、以下に示すいくつかの異なるアプローチが見つかっている。

a. 値レベルのカバレッジ
   - 従来のカバレッジ
   - K 区間被覆性
   - 境界カバレッジ

b. パターンレベルのカバレッジ

c. その他の変種

これらのカバレッジ指標の定義や実験結果については、8.2.3 節で報告している。カバレッジの分析に有効な手法の一つに、 Surprise Adequacy がある。以下では、この手法を我々の問題に適用にしてみる。

Surprise Adequacy に基づくカバレッジ

Surprise Adequacy は、データセットのデータカバレッジや多様性を定義する指標として用いることができる。DSA はテスト入力（x）から最も近い同じラベルの入力（x_a）までの距離と、入力（x_a）から最も近い他のクラスの入力（x_b）までの距離の比である [19]。

実験

距離ベースの Surprise Adequacy（DSA）と Surprise Coverage（SC）の定義に基づき、MNIST データセットを用いてテスト用データセットの DSA の値を算出する実験を行った。これは、テストデータと学習データの類似性と差異を簡潔に示すものである。この実験に使用したモデルを以下にまとめる。

```
ConvNet(
    (layer1): Sequential(
        (0): Conv2d(1, 8, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
        (1): ReLU()
        (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    )
    (layer2): Sequential(
        (0): Conv2d(8, 24, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
        (1): ReLU()
        (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    )
    (drop_out): Dropout(p=0.5, inplace=False)
    (fc1): Linear(in_features=1176, out_features=1000, bias=True)
    (fc2): Linear(in_features=1000, out_features=10, bias=True)
)
```

CNN におけるいくつかの活性化層を対象として Activation Traces（AT）を計算し、その結果を精度に対する DSA 変化としてプロットした。

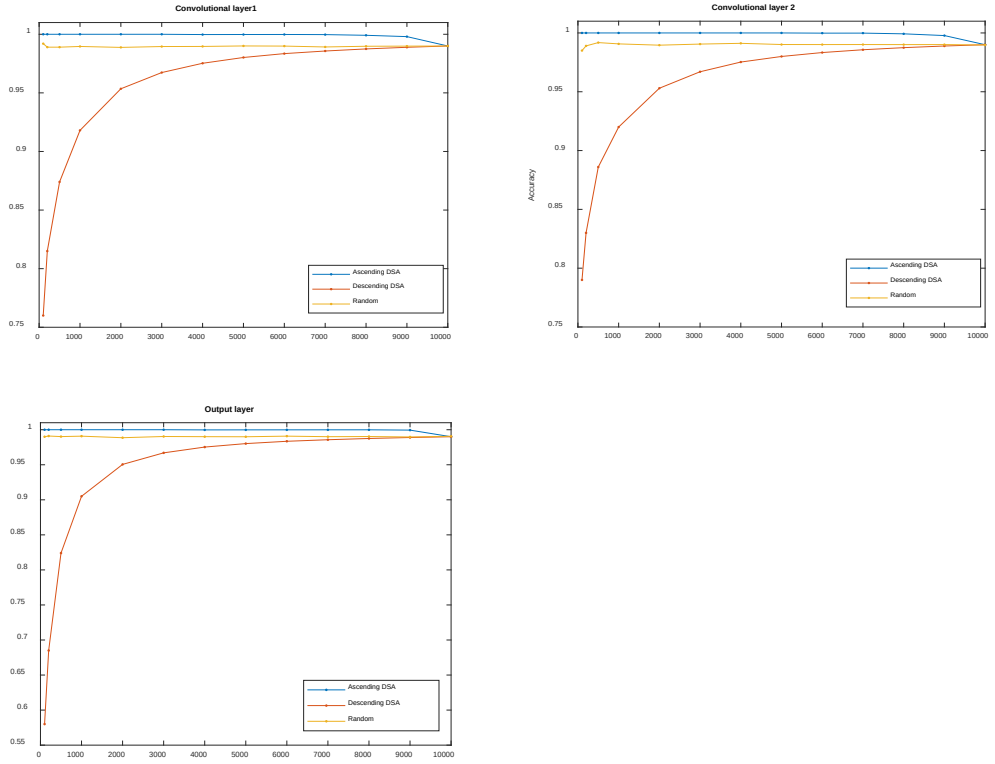


図 76 DSA の変更に伴う精度の変化

ここでは、データを DSA に基づいて（昇順および降順に）並び替えて、グラフを描いた。ネットワークに順次投入することで、DSA とモデルの精度の関係を見ることができる。明らかに、低 surprise データの精度は、高 surprise データより高い。つまり、DSA のスコアが高いほど、モデルが失敗する確率が高くなることがわかる。AI モデルの SC と精度は、ここで扱う問題領域のどの特徴についても計算できる。例えば、数字の**面積**と**長さ**が問題領域の特徴であれば、それらの特徴次元に沿った SC と精度を計算できる。以下に、DSA 分析の結果を示す。

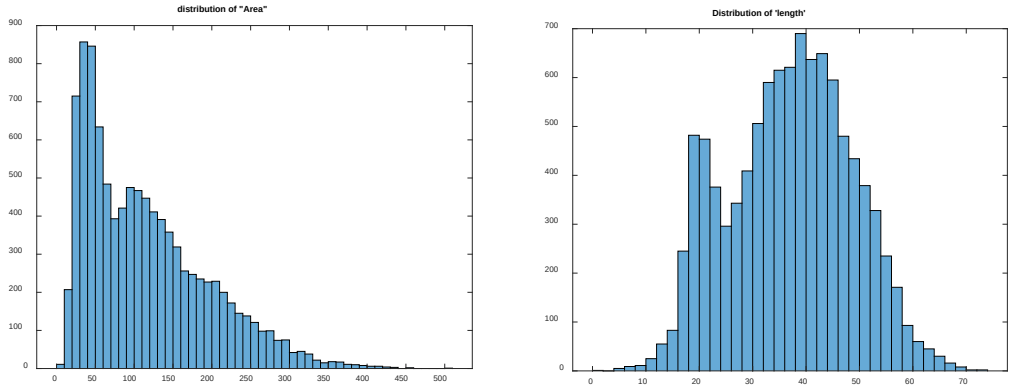


図 77 MNIST (のテスト用データセット) の特徴（面積と長さ）に沿った分布

表 37 特徴「面積」の区分

	小(0, 75)	中(75, 200)	大(200, 500)
データ比率	39.55%	44.48%	15.97%
SC(第 1 層)	0.7438	0.7250	0.5687
SC(第 2 層)	0.7375	0.7312	0.5500
SC(出力層)	0.8063	0.8063	0.6375
精度	0.9901	0.9897	0.9906

表 38 特徴「長さ」の区分

	短い(0, 25)	中程度(25, 50)	長い(50, 75)
データ比率	19.07%	67.24%	13.69%
SC(第 1 層)	0.6750	0.7750	0.5938
SC(第 2 層)	0.6625	0.7813	0.5375
SC(出力層)	0.7000	0.8875	0.6250
精度	0.9911	0.9899	0.9890

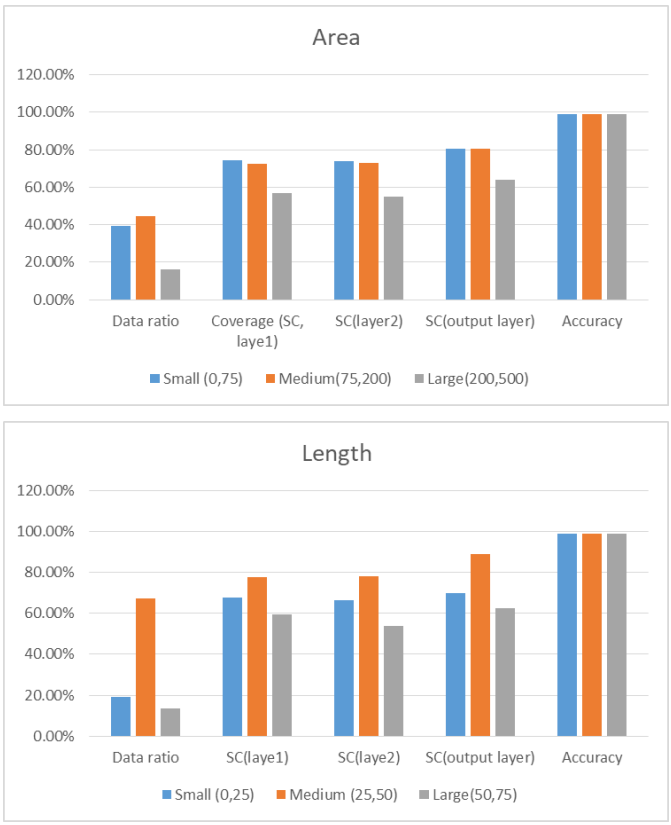


図 78 特徴量「面積」と「長さ」の
各区分の SC と精度

- また、この 2 つの属性を組合せてペアワイズ分析を行うことも可能である。特徴空間の分割を、図 79 に示す。属性ごとの区分の定義により、特徴空間には、特徴の組合せに応じた 9 つのセグメントが存在する。

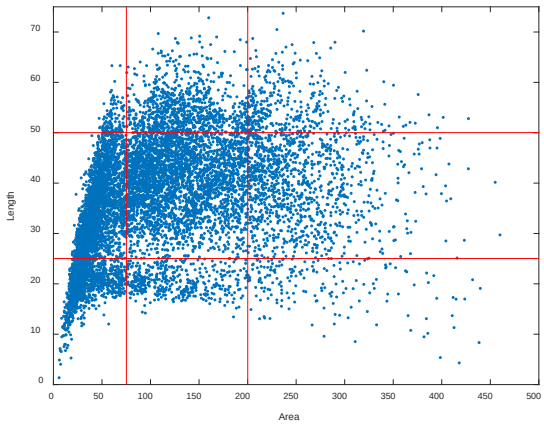


図 79 2 次元(面積、長さ)特徴空間における
データ（MNIST のテストデータ）の分布

そして、各セグメントのデータについて、以下のようにデータ比率、Surprise Coverage

(SC)、精度を算出することができる。

表 39 2 次元問題領域でのデータカバレッジ分析

データ比率		面積		
		小	中	大
長さ	短い	0. 1296	0. 2546	0. 0113
	中程度	0. 0438	0. 3119	0. 0891
	長い	0. 0172	0. 1059	0. 0365
SC(第 1 層)		面積		
		小	中	大
長さ	短い	0. 6500	0. 6500	0. 3500
	中程度	0. 4750	0. 6438	0. 5625
	長い	0. 3813	0. 5375	0. 4438
精度		面積		
		小	中	大
長さ	短い	0. 9907	0. 9898	0. 9912
	中程度	0. 9932	0. 9904	0. 9854
	長い	0. 9942	0. 9887	0. 9973

SC 分析から、モデルの精度とテストデータの SC を関連付けることは困難である。しかし、明らかに SC にはデータ比率と正の相関があり、データ比率は概してデータカバレッジを意味する。そこで、SC をデータカバレッジの尺度として選び、レアケースやコーナーケースの特定に使う。

レアケース・コーナーケースの見極め

カバレッジ分析は、特徴空間を切り分けた各領域におけるデータポイントの必要性に応えるものである。定義によれば、特徴の組合せに対応するデータが存在しない場合に、その組合せを、**高リスクケースと呼ぶ**。現実の世界では

- 高リスクケースが日常的な頻度で発生するなら、データセットの準備に問題がある。
- 高リスクケースが低い頻度で発生する場合は、レアケース/コーナーケースとして扱う。

例えば、上記のカバレッジ計算から、巨大な数字の画像は、実運用で現れる可能性があるにも関わらず、訓練用データセットには存在しないことが分かった。つまり、巨大数字の画像はコーナーケースである。

DSA に基づくコーナーケース検出

SA はテストデータと訓練用データセットの間の新規性を表現することができる。個々のテストデータポイントについて、SA は訓練データ全体との違いや類似度を表す。したがって、SA はコーナーケースを捉えるのに有効な指標であると考えられる。コーナーケース

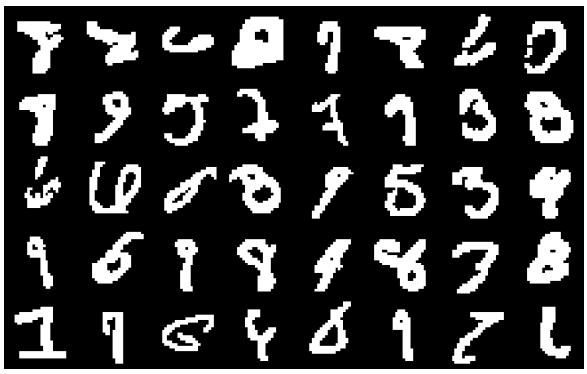
の集合を以下のように定義する。

$$\boldsymbol{Corner\ cases}: \{\boldsymbol{x}|\boldsymbol{class}(\boldsymbol{x}+\boldsymbol{pert}) \neq \boldsymbol{class}(\boldsymbol{x}), |\boldsymbol{pert}| < \boldsymbol{\varepsilon}\} \quad (17)$$

ここでも、DSA を測定値として使用し、DSA の変化に対する精度の変化を得ることができる。そして、SA とコーナーケースの関係をさらに分析することができる。

先に説明した CNN を使用した。テスト精度は 99%であり、誤って分類された画像は全部で 100 枚である。図 76 の精度対 DSA のグラフを見ると、すべての外れ値で DSA が高いわけではなく、正しい画像でも DSA が高いものがあると分かる。高い DSA 値に対して最も精度が低いのは、出力層に基づく DSA ランキングなので、これを使うと最も多くの誤分類される画像をコーナーケースとして検出することができる。以下の入力画像は、出力層に基づく DSA が最も大きい画像である。

表 40 DSA が最も大きい画像

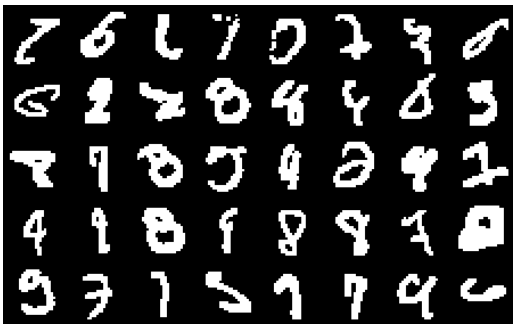
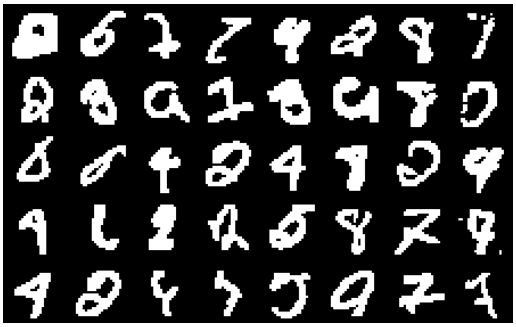
画像	ラベル
	実際のラベル [8 2 6 2 7 8 6 0 8 9 5 7 7 7 3 8 6 6 0 8 9 5 3 4 9 5 9 8 4 8 3 8 1 7 6 9 0 9 7 6] 推定したラベル（13） [7 7 4 0 9 7 6 7 7 9 7 2 3 9 5 8 6 6 8 0 9 5 3 9 1 6 9 9 4 8 7 8 1 9 5 4 6 1 8 1]

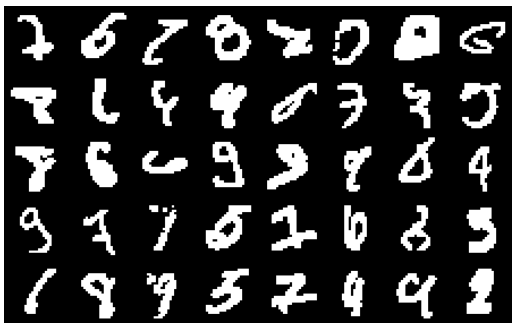
これらの高 DSA 入力画像のうち、27 枚の画像で推定が間違っており、このモデルのコーナーケースとなっている。しかし、正しく推定した画像も 13 枚あり、コーナーケースを特定するよりよい指標が必要である。

DSA の計算には、3 つの変更案 [42]がある。DSA に基づくコーナーケース検出指標を改善するために、これらの変更方式を用い、その結果と出力を比較し、ここでの問題空間に最適な DSA 指標を選択した。

これらの新しい DSA の定義を適用し、DSA が高い入力に対するモデル予測から、各変更方式によるコーナーケースデータを得た。以下の表は、3 つの変更 DSA のコーナーケース検出性能を比較したものである。なお、Activation Trace の計算には、出力層のみを使用した。

表 41 DSA1、DSA2、または DSA3 が大きい画像

	画像	ラベル
DSA1		正しいラベル [7 5 6 7 0 7 3 0 6 2 2 8 8 9 0 5 8 7 8 5 9 2 4 1 4 9 8 9 8 8 7 2 9 7 7 5 7 7 9 6] 推定したラベル (8) [8 6 1 1 7 2 7 8 5 8 7 0 8 4 6 3 7 9 0 7 4 2 9 2 9 9 8 9 8 9 3 0 5 3 1 5 9 7 4 4]
DSA2		正しいラベル [2 5 7 7 4 2 8 7 2 3 9 1 3 9 8 0 0 0 4 2 4 8 2 4 4 6 2 2 5 8 7 7 4 2 9 7 5 9 7 7] 推定したラベル (9) [0 6 2 8 9 8 9 1 8 8 9 2 8 9 7 7 6 8 9 2 4 7 0 9 9 1 8 2 0 8 2 7 4 0 4 7 7 0 2 3]

DSA3		正しいラベル [7 5 7 8 2 0 2 6 8 6 9 4 0 7 3 5 8 6 6 9 9 9 0 4 9 7 7 5 1 6 8 5 6 8 9 5 7 9 9 2] 推定したラベル(3) [2 6 8 0 7 7 0 5 7 1 4 9 8 3 7 7 7 6 4 3 3 8 6 9 5 3 1 0 2 6 2 3 1 9 7 5 2 4 4 8]

この結果から、モデルが誤った挙動を示す入力を特定することについて、DSA3 が最も成功していることがわかる。つまり、敵対的なデータに対して DSA3 の定義を使用することで、最も多数のコーナーケースを得ることができる。

特徴削減

カバレッジの結果に基づいて、適切なデータ設計を行う必要がある。その際、問題領域から除外できる特徴や特徴値の範囲が見つかるかもしれない。

例えば、データセット構造に対するモデルの挙動を知るために、特徴ごとのデータ変更テストを行うことができる。出力が変化しなければ、その変更はモデルの性能に影響を与えないことを意味する。したがって、特徴空間からその特徴または特徴値を省くことができる。

おわりに

この MLQM 基準はデータを評価するためのもので、問題領域に対するデータカバレッジに関するものである。この分析に基づいて、訓練用データセットから訓練能力を、テスト用データセットから性能測定能力を判定できる。データカバレッジを改善するために、レアケースのデータの補充に使える方法を後の節で述べる。次の節では、データの分布、すなわち、データセットの均一性（偏りのなさ）について分析する。

9.5.4 B-2: データセットの均一性

定義

この MLQM 基準は、対象とする特徴空間におけるデータセットの分布の計算を対象とする。被覆性分析が特徴空間の隅々でのデータの有無を問うのに対し、均一性は、それらの領域におけるデータの密度を分析する。本節では、データセット分布の計算と可視化を行い、その均一性を判断する。

視覚的な表現からデータ分布を把握する

例えば、視覚的な観点から、特徴「**太さ**」に関するデータ分布を分析したいとしよう。

画像中の手書き数字の太さを測定するのは難しい。しかし、太い数字のピクセル数が多いことを考慮すれば、太さを示す数値の分布をおおよそ推定することができる。そのためには、バウンディングボックスの代わりに輪郭線を描くことを考えればよい。

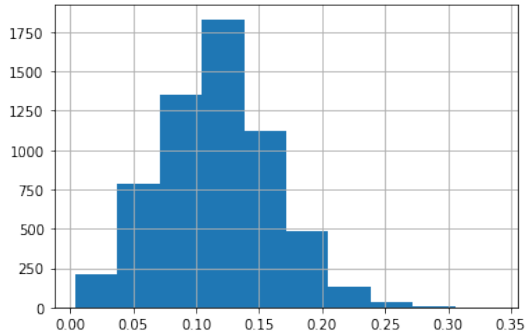


図 80 数字の太さに基づくデータ分布

図 80 の分布図において、X 軸は数字を構成するピクセルの比率を表している。つまり、20%までは x 軸に沿った正規分布があることがわかる。完全に一様な分布ではないが、このデータセットには、ここで扱う**太さ**の範囲について、十分な量のデータがある。

データの均等性を計算する

上記の被覆性分析から、データの均等性を評価するための簡単な指標を提案することができた。TPCov の考え方を用いて、均一性を計算することができる。

均等性を考慮する場合、上記の被覆性を元に新たな指標を作ることができる。例えば、値に基づく被覆性指標として最も簡単なのは上位 p%被覆性(TPCov)で、最も密度の高いデータポイントの p%がある領域と元の被覆範囲の比率として定義でき、以下のように表せる。

$$\boldsymbol{TPCov}[\boldsymbol{x(n)}] = \frac{|\{\boldsymbol{S}|\boldsymbol{density(S)} = \boldsymbol{p}\%\}|}{\boldsymbol{high_n - low_n}} \tag{18}$$

ここで、S は上位 p%のデータが被覆する範囲である。

さて、TPCov を用いると均等性を評価する指標は、TPCov と p%の差として、次のように定義される。

$$\boldsymbol{EI_{err}} = |\boldsymbol{TPCov} - \boldsymbol{p}\%| \tag{19}$$

図 81 に示すように、EI_{err} の値が低いならば、データが均等に分布していることを意味する。

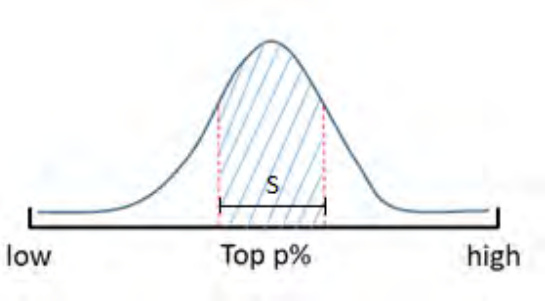


図 81 均等性計算の説明のための
データ分布の概念図

p%の異なる値を考慮すれば、さらに異なる p%に対する被覆性指標の値を比較することも考えられる。その場合、均等性を測定するために、次のように定義した Area under the curve（AUC）を見ることができる。

$$\boldsymbol{AUC = area\{coverage(p): p \in (0、\ 100\%)\} = \int_0^1 Coverage(p)dp} \qquad (20)$$

データ収集の偏りを減らす

レアケース密度を高めようとすると、データ収集の手順に偏りが生じる。すると、ML モデルの平均的パフォーマンスを損なう恐れがある。したがって、データ収集における偏りのレベルを最適化する必要がある。

データ設計

データの評価に関する最後の内部品質に関する節を終えるにあたり、データ品質を向上させる方法についての研究、すなわち様々なデータ設計プロセスを説明する必要がある。データ設計とは、対象とする問題領域を念頭に置いて、機械学習モデル用のデータセットを準備することである。上記の問題領域は特徴空間と言い換えることができる。このパートでは、特徴空間の各領域に対して十分なデータポイントを作成あるいは収集するためのいくつかの方法について述べる。

次の 2 通りの方法がある。

- 新しいデータセットを構築する
- 既存のデータセットを加工する

データ設計の理由

データ設計は普通、AI 開発者の仕事とみなされるが、次のような例では、AI 評価者がデータ設計をする必要があるかもしれない。

– 例えば、開発者の要求分析と評価者の要求分析が異なるとする。その場合、評価者は評価のための適切なテスト用データセットを設計する必要がある。
– 一方、開発者と評価者が同じ要件で作業する場合、評価者は開発者のデータセット構築

プロセスを自分の評価に採用することができる。

新しいデータセットを構築する

十分な人手や技術があれば、ゼロからデータセットを構築できる。その場合、以下のようになる。

– すでに説明したすべての特徴を含むことを容易に保証できる。
– また、各ケースに含まれるデータポイントの数を設定できる。
– それにより、後の**被覆性**や**均一性**の分析は、単にグラフ表現を用意するだけで済む。

既存のデータセットで作業する

新しいデータセットの構築は、今回はスコープ外である。そこで、ここでは、MNIST という最も有名な手書き数字分類のデータセットを採用して、MLQM のワークフローを実証することにする。さて、あらかじめ定義されたデータセットを利用する場合の課題として

– すべての特徴次元のデータがあるとは期待できない。後節で被覆性を確認する必要がある。
– また、特徴空間全体に一様な分布を確保することはできない。
– その後、**被覆性**と**均一性**分析において、データセットの実際の分布を調べ、データ拡張が必要なケースを特定することができる。

データ設計の手順

既存のデータセットを問題に利用する場合、定義された特徴空間にうまく分散したデータポイントを得ることはできない。そこで、本節では、データを拡張したり、データカバレッジを高めるための方法論をいくつか定義する。

データ拡張

データ拡張の方法については、特徴空間の各領域の被覆率に着目して詳細に説明する。例えば、ここでは、明度/コントラスト特徴量に対する拡張処理の候補を一つ紹介する。

- **コントラストを変化させることでデータを拡張する**　これは、MNIST データセットのデータポイントがほとんど存在しない特徴次元の 1 つである。この特徴量に対して新しいデータポイントを追加するには、単に**与えられたデータの一部を暗くすればよい**。以下にその例を示す。

外部処理追加による特徴削減

特徴量次元によっては、データカバレッジを高めるための適切なデータ拡張法が見つからないことがある。そこで、そのような特徴量に対応する外部処理を追加することも考えられる。これにより、その特徴を問題領域から除外することができ、特徴の数を減らすことができるだけでなく、訓練用データやテスト用データの数も減らすことができる。例えば、任意の画像フレーム中で数字を中央に寄せるための外部処理についてはすでに述べた。

外部処理による数字の位置の処理　MNIST のような反転画像では、数字のバウンディングボックスを簡単に得ることができる。そして、その枠を枠の中央に配置することで、**数字の位置**に関する要件をなくすことができる。以下は、MNIST のテストセットからの例である。

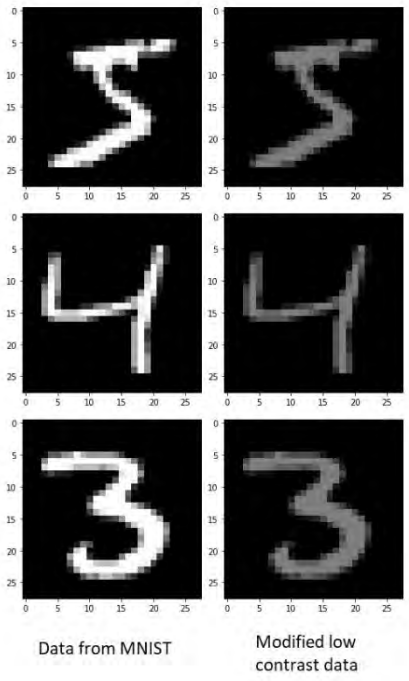


図 82　コントラストの変更

機械学習はデータドリブンのプロセスであるため、適切なデータ管理が必須となる。本節では、そのためのいくつかの手順を説明したが、同様の方法では扱えない特徴もあり得る。そのような特徴については、手動でデータセットを構築することが選択肢となる。それも不可能であれば、開発者はその特徴に対応したモデルを訓練できず、評価者も評価のための十分なテスト用セットを構築することができなくなる。その場合、その特徴は ML モデルのスコープ外とせざるを得ない。

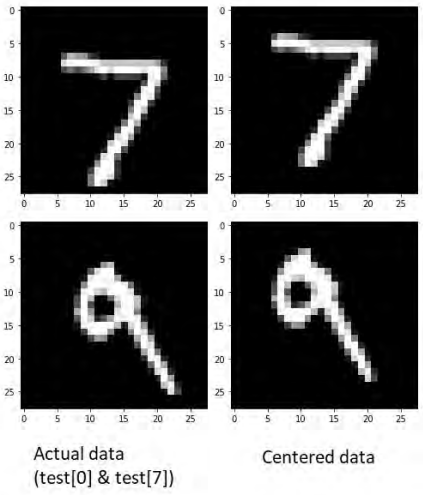


図 83　画像を中央に寄せる

おわりに

均一性は、テスト用データよりも訓練用データにおいてより重要な基準である。これは本質的に、機械学習モデルの出力の偏りに関わっている。この基準を満たすことで、モデルはより高い性能、コーナーケースに対する精度、リスク要因の回避を達成することができる。

9.5.5 B-3: データの妥当性

この版のリファレンスガイドでは、本内部品質を検討していない。

9.5.6 C-1: 機械学習モデルの正確性

定義

精度は、モデルの性能評価に使う、モデルの正確性の主な尺度である。以下の節では、正確性の測定に使える基準または KPI をいくつか定義し、訓練済みモデルの出力への適用例を示す。

さまざまな精度指標と KPI

郵便番号分析は分類問題であるため、一般的に使用される性能指標は混同行列である。この行列は以下のように可視化できる。

表 42 2 値分類の混同行列

		予測した出力	
		正	負
正しい出力	正	TP	FN
	負	FP	TN

混同行列　これは、2 値分類のための単純な混同行列である。4 種類の出力動作の可能性がある。

TP = 予測器が正しく正を予測した場合
FP = 予測器が誤って正を予測した場合
FN = 予測器が誤って負を予測した場合
TN = 予測器が正しく負を予測した場合

この行列に基づき、有用かつよく使われる性能指標をいくつか定義することができる。

精度　これは、モデルによる正しい予測の指標であり、次のように定義できる。

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP} + \boldsymbol{TN}}{\boldsymbol{TP} + \boldsymbol{TN} + \boldsymbol{FP} + \boldsymbol{FN}} \tag{21}$$

再現率　これは、正のデータの総数に対する正しい正の予測の比率であり、次のように定義できる。

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FN}} \tag{22}$$

適合率　これは、予測された正の総数に対する正しい正の予測の比率であり、次のように定義することができる。

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FP}} \tag{23}$$

F スコア　これは、適合率と再現率の調和平均であり、次のように定義することができる。

$$\boldsymbol{F - measure} = \frac{\boldsymbol{2 \times Recall \times Precision}}{\boldsymbol{Recall + Precision}} \tag{24}$$

例えば、**郵便番号の分析**問題のために、MNIST データセットで訓練する CNN を選択した。モデルの構成は以下の通りである。

表 43 郵便番号の分析のモデル

アーキテクチャ	Conv(24,24,24)+ReLU MaxPooling(12,12) Conv(8,8,64)+ReLU MaxPooling(4,4) Flatten() FC(1000)+ReLU FC(10)+Softmax
訓練可能なパラメータ数	1,074,098
訓練精度	99.26%
テスト精度	99.50%

精度以外の KPI 指標は、テスト用セットを使ってすべてのクラスについて個別に計算し、以下の表に示す。

表 44 全クラスの KPI

	0	1	2	3	4	5	6	7	8	9
再現率	0.998	0.999	0.998	0.998	0.995	0.990	0.987	0.992	0.996	0.995
適合率	0.994	0.996	0.995	0.994	0.994	0.994	0.999	0.997	0.995	0.991
F スコア	0.996	0.998	0.997	0.996	0.994	0.992	0.993	0.995	0.995	0.993

訓練済みモデルのテスト精度は非常に良いが、再現率、適合率、F スコアを見れば、モデルのより深い挙動を知ることができる。再現率の結果を見ると、この分類器は「1」の予測で最もよく、「6」の予測で最も悪い。適合率の結果では、この分類器は「6」の予測で最も間違いが少なく、「9」の予測を最も多く間違うことが分かる。F スコアの結果から、全体的なパフォーマンスは、数字の「1」に対して最も良く、数字の「5」に対して悪いと言える。

モデルの動作定義とコーナーケースの検出

これまで、コーナーケースを見つけるには、データカバレッジやデータ分布を見て判断していた。しかし、コーナーケースは、テスト用データに対するモデルの挙動からも特定することができる。あるモデルが間違った予測をする入力データを、そのモデルに関する未同定事例と呼ぶことができる。類似のモデルが誤った予測をする入力データを分離できれば、その特定のソリューションに関するコーナーケースが得られる。モデルが誤った挙動を示す入力データを優先して扱うことに関する研究もある [43].

おわりに

精度はモデルのパフォーマンスの一面でしかないが、その中でも、様々な観点からモデ

ルを評価することになる。例えば、正例と負例を別々に識別する精度や、モデルの挙動に基づくコーナーケースの識別などである。次節では、ML モデルの頑健性、すなわち安定性を定義し評価する。

9.5.7 C-2: 機械学習モデルの安定性

定義

安定性は ML モデルの最も重要な特性の一つであり、入力データとモデルの両方に摂動を与えた場合のモデルの挙動を決定する。安定性は、従来から次のように定義されている[34]。「システムまたは構成要素が、無効な入力やストレスの多い環境条件の存在下で、正しく機能する度合い。」

S を機械学習システムであるとする。S の正しさを E(S)とする。データ、学習プログラム、フレームワークなど、任意の機械学習要素に摂動を与えた機械学習システムを δ(S)とする。機械学習システムの頑健性は、E(S)と E(δ(S))の差の測定値である。

$$\boldsymbol{r = E(S) - E\big(\delta(S)\big)} \tag{25}$$

このように、頑健性とは、ML システムの正しさが摂動の下で回復する力を測定するものである。ここでは、いくつかの頑健性の指標を定義し、**郵便番号の分析**における実験結果を示す。

頑健性は、AI を構成する 2 つの基本要素、すなわちデータとモデルについて定義することができる。以下では、この 2 つの指標を別々に考えてみた。

データの摂動に対する頑健性

データに関する頑健性の指標は、敵対的データに依存する。敵対的データは、テスト用データの近傍に作成することができる。元のデータと敵対的データとのベクトル距離が頑健性の指標とみなされる。頑健性/安定性を計算するための様々な測定基準がある。

- **局所的敵対的頑健性(Local Adversarial Robustness)** x を ML モデル h のテスト用入力とする。x'を x に対する敵対的摂動によって生成された別のテスト用入力とする。モデル h は、任意の x'に対して以下を満たすならば、入力 x において δ-局所頑健である。

  $$\forall \boldsymbol{x'}: \big|\big|\boldsymbol{x - x'}\big|\big|_{\boldsymbol{p}} = \boldsymbol{\delta} \rightarrow \boldsymbol{h(x) = h(x')} \tag{26}$$

  $\big|\big|*\big|\big|_p$ は距離測定の p ノルムを表す。機械学習のテストでよく使われる p は 0、1、2 である。

- **大域的敵対的頑健性(Global Adversarial Robustness)** x を ML モデル h のテスト用入力とする。x'を x に対する敵対的摂動によって生成された別のテスト用入力とする。モデル h は、任意の x と x'に対して以下を満たすならば、δ-大域頑健である。

$$\forall \boldsymbol{x}、\ \boldsymbol{x'}: \left|\left|\boldsymbol{x}-\boldsymbol{x'}\right|\right|_{\boldsymbol{p}} = \boldsymbol{\delta} \rightarrow \boldsymbol{h}(\boldsymbol{x}) - \boldsymbol{h}(\boldsymbol{x'}) \leq \boldsymbol{\epsilon} \tag{27}$$

上記の敵対的頑健性の定義に基づけば、δ の値を頑健性の測定値として直接利用することが可能である。しかし、異なるモデルの頑健性を評価するためには、これらの頑健性指標を相対的な指標に一般化する必要がある場合がある。ここで、入力データが[0、1]^d に正規化されており、d は次元数であると仮定すると、相対的な頑健性指標は以下のように定義できる。

$$\boldsymbol{r_1} = \frac{\boldsymbol{\delta_1}}{\boldsymbol{0.5 \times d}}, \quad \left|\left|\boldsymbol{x}-\boldsymbol{x'}\right|\right|_{\boldsymbol{1}} = \boldsymbol{\delta_1} \tag{28}$$

$$\boldsymbol{r_2} = \frac{\boldsymbol{\delta_2}}{\sqrt{\boldsymbol{0.5 \times d}}}, \quad \left|\left|\boldsymbol{x}-\boldsymbol{x'}\right|\right|_{\boldsymbol{2}} = \boldsymbol{\delta_2} \tag{29}$$

$$\boldsymbol{r_\infty} = \frac{\boldsymbol{\delta_\infty}}{\boldsymbol{0.5}}, \quad \left|\left|\boldsymbol{x}-\boldsymbol{x'}\right|\right|_{\boldsymbol{\infty}} = \boldsymbol{\delta_\infty} \tag{30}$$

頑健性(δ)の測定については、いくつかの研究、例えば、CNN-Cert [44]および Fast-Lin [45]がある。

測定された δ 値を用いて、上記で定式化された相対頑健性を算出することができる。例として、MNIST のデータを使い、CNN-Cert と Fast-Lin の結果を用いて相対頑健性の指標を算出したものを示す。

表 45 相対頑健性指標

		検証した下限値(δ)		相対頑健性 (r)(×10^{-2})	
	L_p ノルム	CNN-Cert	Fast-Lin	CNN-Cert	Fast-Lin
MNIST 4 層 5 フィルター 8680 隠れノード	L_∞ L_2 L_1	0.0491 0.1793 0.3363	0.0406 0.1453 0.2764	9.82 0.91 8.58	8.12 0.73 7.05
MNIST 4 層 20 フィルター 34720 隠れノード	L_∞ L_2 L_1	0.0340 0.1242 0.2404	0.0291 0.1039 0.1993	6.80 0.63 6.13	5.82 0.52 5.08
MNIST 5 層 5 フィルター 10680 隠れノード	L_∞ L_2 L_1	0.0305 0.1262 0.2482	0.0248 0.1007 0.2013	6.10 0.64 6.33	4.96 0.51 5.14

モデルの摂動に対する頑健性

モデルに関する頑健性の指標はモデルへの摂動に依存する。モデルレベルの頑健性を計

算する様々な方法がある。

- **パラメータの頑健性**　w は ML モデル h のパラメータとする。w'は w に若干の摂動を加えて作った別のパラメータとする。モデル h は、任意の w' に対して以下を満たすとき、パラメータ摂動に対して δ_w 局所頑健である。

$$\forall \boldsymbol{w}': \left|\left|\boldsymbol{w}-\boldsymbol{w}'\right|\right|_{\boldsymbol{p}} = \boldsymbol{\delta_w} \rightarrow \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{h}'(\boldsymbol{x}) \tag{31}$$

図 84 は、AI モデルの頑健性評価におけるパラメータ摂動の例である。頑健性の指標として最小距離 δ_{min} を求めることを目的とした敵対的頑健性に比べ、パラメータ頑健性は最大距離 $\delta_{w\ max}$ を頑健性の指標として利用する。また、プログラムレベルの頑健性は、敵対的データの生成やその認証が不要であるという利点がある。

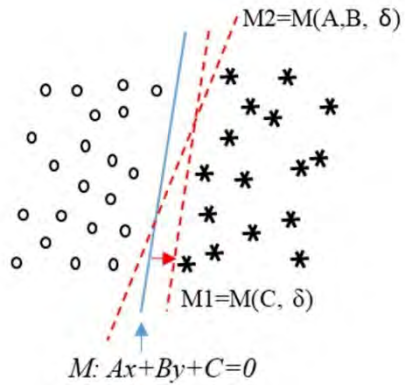


図 84 パラメータ摂動による線形分類器の変化

- **変異頑健性**　ソフトウェアの変異に対する頑健性を測定するために、ソフトウェアプログラム P (すべてのソフトウェアプログラムの集合 $\mathcal{P}$ のメンバー)、変異演算子の集合 M (ここで各 $m \in$ M は関数写像$\mathcal{P} \rightarrow \mathcal{P}$)、およびテストスイート $T: \mathcal{P} \rightarrow$ {true, false}に関してそれを定式化する。プログラム P は、T(P) = true のときのみ、テストスイートに合格するという。プログラム P、変異演算子の集合 M、および T(P) = 真となるテストスイート T が与えられたとき、**ソフトウェア変異頑健性**を *MutRB*(P, T, M) と表記し、コンパイルできて T に合格するすべての直接突然変異 P' = m(P), $\forall$ m∈M の割合とする。

$$\boldsymbol{MutRB}(\boldsymbol{P},\boldsymbol{T},\boldsymbol{M}) = \frac{|\{\boldsymbol{P}' \mid \boldsymbol{m} \in \boldsymbol{M}.\boldsymbol{P}' = \boldsymbol{m}(\boldsymbol{P}) \cap \boldsymbol{T}(\boldsymbol{P}') = \boldsymbol{true}\}|}{|\{\boldsymbol{P}' | \boldsymbol{m} \in \boldsymbol{M}.\boldsymbol{P}' = \boldsymbol{m}(\boldsymbol{P})\}|} \tag{32}$$

この測定は、MLQM に利用できる。例えば、DeepMutation [46]の結果を例にとると、ここでは、3 つの DL モデル A, B, C が MNIST データセットでテストされている。

表 46 変異頑健性

モデル A	モデル B	モデル C

アーキテクチャ	Conv(6,5,5)+ReLU MaxPooling (2,2) Conv(16,5,5)+ReLU MaxPooling (2,2) Flatten() FC(120)+ReLU FC(84)+ReLU FC(10)+Softmax	Conv(32,3,3)+ReLU Conv(32,3,3)+ReLU MaxPooling(2,2) Conv(64,3,3)+ReLU Conv(64,3,3)+ReLU MaxPooling(2,2) Flatten() FC(200)+ReLU FC(10)+Softmax	Conv(32,3,3)+ReLU Conv(32,3,3)+ReLU MaxPooling(2,2) Conv(64,3,3)+ReLU Conv(64,3,3)+ReLU Maxpooling(2,2) Flatten() FC(200)+ReLU FC(200)+ReLU FC(10)+Softmax
訓練可能パラメータ	107,786	694,402	698,402
訓練精度	97.40%	99.30%	99.50%

各変異演算子について、20 個の DL 変異体を作成し、変異スコアを算出した。DeepMutation の変異スコア(mutation score)と MutRB は相補的で、Mutation_score + MutRB=1 である。以下、文献で紹介されている変異スコアと、計算した MutRB スコアを別々の表で紹介する。

表 47 変異スコア(%)

	0	1	2	3	4	5	6	7	8	9
モデル A	7.22	8.75	9.03	6.25	8.75	8.19	8.75	9.17	9.72	9.03
モデル B	1.59	3.29	8.29	7.44	5.49	4.02	8.17	3.66	5.85	8.41
モデル C	8.33	7.95	8.97	9.74	9.74	9.62	9.62	8.97	9.74	7.56

表 48 MutRB スコア(%)

	0	1	2	3	4	5	6	7	8	9
モデル A	92.78	91.25	90.97	93.75	91.25	91.81	91.25	90.83	90.28	90.97
モデル B	98.41	96.71	91.71	92.56	94.51	95.98	91.83	96.34	94.15	91.59
モデル C	91.67	92.05	91.03	90.26	90.26	90.38	90.38	91.03	90.26	92.44

モデルの失敗を狙った敵対的データを作成することに焦点を当てた研究がいくつかある。入力データにある程度の摂動を加えることにより、これらの研究を頑健性測定に採用することも可能である。以下に、そのような研究のいくつかを挙げておく。

- Deepxplore: Automated Whitebox Testing of Deep Learning Systems [28]
- Guiding Deep Learning System Testing Using Surprise Adequacy [19]
- TensorFuzz: Debugging Neural Networks with Coverage-Guided Fuzzing [47]

おわりに

頑健性の指標は、未知の入力や新しい環境条件に対するモデルのパフォーマンスを示す。

ここで定義した頑健性の KPI により、郵便番号分析のための訓練済みモデルが、クライアントやユーザーが設定した安定性の要求レベルを満たしているかどうかを確認できる。また、これは AI ソリューションを評価する最後の内部品質である。以降の MLQM の内部特性は AI システムの他の部分を評価するためのものである。他の部分は、AI が公共の場や制御された環境で動作する前またはその最中に、その AI をサポートする。

9.5.8 D-1：プログラムの信頼性

定義

MLQM ガイドラインによると、**プログラムの信頼性**とは、機械学習の訓練段階や予測やインターフェースに用いられるソフトウェア部品が、学習データおよび学習済み ML モデルに対応してそれぞれ実行されたときに、正しく動作することを意味する。AI ソリューションはゼロから作られることは少ない。多くの場合、画像処理ソフトウェア、Python のパッケージやライブラリなど、多数のソフトウェア部品から構成されている。以下のような AI 要素を詳細に記述し、その品質を検証する必要がある。

プログラム&オープンソースライブラリ

この AI の開発には、Python 言語が使用されている。また、様々なオープンソースパッケージを使用しており、互いにバージョン互換性があることが望ましい。そのため、使用するパッケージとそのバージョンの一覧を開発者が示す必要がある。またパッケージでなくても、オープンソースでなくても、利用したプログラムについては同様にバージョンを記載することが望ましい。

表 49 使用したパッケージとそのバージョンの一覧

プログラミング言語	バージョン
Python	3.6.12
パッケージ	バージョン
NumPy	1.18.5
TensorFlow	2.3.1
Pillow (PIL fork)	8.0.1

画像処理ユニット

これは、カメラでの撮影や、データセット同様の入力画像の作成処理を指す。

簡単のために、郵便番号の撮影は 3 チャンネル（RGB）の 2 次元画像を生成するカメラで行う。これらの画像は、訓練済み AI モデルに適した入力に変換するアルゴリズムやプログラムフローの対象となる。

例として、白い紙に黒いインクで書かれた数字(6)の一般的な画像を撮影した。この画像を、MNIST の手書き数字データセットと同様の白黒（28, 28）画像に変換するプログラムを作成した。

データの前処理には Python を使用し、画像処理ツールにはオープンソースパッケージの Pillow を使用した。

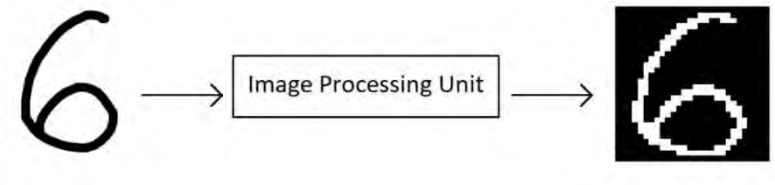


図 85 画像処理ユニットによる画像の修正

外部処理用ユニット

このデバイスのこの部分では、9.5.2 節「データ設計の十分性」で説明した、いくつかの問題領域特性を省略するための外部処理のアルゴリズムを定義する。このアルゴリズムの正しさを検証する必要がある。すでに、定義された特徴である**数字の位置**を削除する外部処理を説明した。この処理のために開発したアルゴリズムは、データの前処理の後ろ、モデルへの入力の前に置く。

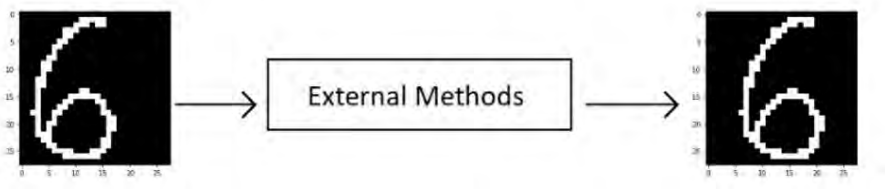


図 86 外部処理ユニットによる画像の修正

このユニットは、入力画像に対して順次実行される 1 つまたは複数の外部処理で構成できる。**数字の位置**を処理するアルゴリズムは、オープンソースパッケージである NumPy と Pillow の助けを借りて、プログラミング言語 Python を使って書いた。

メモリの使用状況

AI デバイスの動作中のメモリの最小使用量と最大使用量を定義する必要がある。

- **モデルのアーキテクチャと重み**　訓練済み AI ネットワークとその重みを保存する際に使うフォーマットの 1 つに HDF (Hierarchical Data Format) ファイル (.h5) がある。9.5.6 節「機械学習モデルの正確性」では、訓練済み CNN の結果を紹介した。そこで保存したネットワークは 1,074,098 個のパラメータを持ち、ハードディスクに約 12.3MB の容量を必要とする。
- **入力データ**　入力データとは、カメラの画像データ、切り詰めた画像、前処理や外部処理で変換した画像などを指す。変換後の画像サイズを量ってもよいが、ハードディスク上で最大容量を占めるのは元の画像である。そのため、入力データによるメモリ使用量は、装置、カメラ、品質、解像度などを完全に設計した後に定義することができる。
- **コード／アルゴリズム**　プログラミング言語によって書かれたさまざまなアルゴリズムが、装置のワークフローの一部となっている。これらのコードはハードディスクにそ

れほど大きな容量を必要としない。例えば、前述した外部処理のアルゴリズムが必要とする容量は約 4KB である。

- **再訓練用データセット**　再訓練用データセットを保持するために、少なくとも実際のデータセットと同じ大きさのスペースを確保する必要がある。例えば、MNIST データセットが必要とする容量は約 52.4MB である。
- **ネットワークの訓練用の RAM と GPU**　運用の合間にモデルを訓練する必要がある場合、その処理のためのマシンスペックを定義する必要がある。例えば、ここで説明した MNIST データセットを用いたモデルの訓練は、8GB の RAM、GPU なしで、妥当な時間で行うことができる。

メモリや装置に関する全ての要件を総合することで、デバイスのメモリ割り当てを設計することができる。

時間コスト

実世界に適用する際には、時間は課題となる。時間がかかるほど、装置の効率が低下する。私たちは、装置のどの段階でも無駄な時間ロスをなくすと同時に、アルゴリズムをより高速に改善する方向性を示す必要がある。

例えば、郵便番号分析では、ワークフロー全体の中で最も時間がかかるのは分類処理であり、ここで計算の大部分が行われる。そのため、バッチ実行の方が逐次実行よりも高速で効率的だが、その分、必要なメモリ容量が大きくなる。

ソフトウェアのセキュリティ

装置がオンラインで動作する場合、セキュリティは重要な問題である。AI・機械学習機能を持つアプリケーションセットを構築する際に、ソリューション設計者が考慮すべき事項については、ガイドラインのセキュリティに関する章（第 2 版では 9 章）を参照されたい。

訓練環境と運用環境の違い

特に機械学習の応用では、訓練環境と実運用環境が異なることが多く、数値計算の挙動が変化することも少なくない。このような場合、両環境の違いを評価し、それによって装置がもたらす結果がどのように変わるかを定義する必要がある。

例えば、ML モデルは訓練によってランダムな初期値から最適化された損失関数に至る。この過程は、装置が暗黙のうちに行う多数の計算から構成されている。そのため、装置によって KPI の結果が異なることはよくあることである。そのため、様々な装置でシステム全体を評価し、装置環境の変化によるパフォーマンスの変動範囲を見積もる必要がある。

おわりに

上記のプログラム要素は、ソリューションが運用段階に入る前に定義し、品質とセキュリティを検証して確保しなければならない。

9.5.9 E-1：運用時品質の維持性

定義

機械はとても信頼性が高いが、故障することもある。そのため、メンテナンスは定期的な作業であり、稼働中のすべての機械につきものである。MLQM ガイドラインでは、「運転開始時に満たした内部品質を運転期間中維持するための技術を記述する」としている。

メンテナンス時の作業フロー

ここでは、稼働中の装置のさまざまな不具合と、それを克服するための手順について説明する。運用中に起こりうる危機を例示するために、制限された特徴次元に基づいて加工したデータセットを使用する。特徴量**面積**を特徴次元の一つとして考えると、トレーニングセットとテストセットでは、分布が異なり、値の範囲も異なることがわかる。

図 87 の通り、訓練用セットの**面積**の範囲は 22.75〜315.125 であるのに対し、テスト用セットでは 5.75〜505.25 である。訓練用セットが問題領域の範囲を定義していると考えると、テスト用セットも同じ範囲の値を持つことになる。テスト用データセットからサブセットを取り出して、期待されるテスト用セット（test_original1）を表すものと、運用中の定義範囲外の入力を表すテスト用データセット（test_operational1）を用意した。

特徴量**コントラスト**についても考慮する。テスト用セット（test_original2）の画像はすべて高コントラストであるため、データ設計手順の 1 つを使用して、ありそうな運用中入力を表す類似のテスト用データセット（10,000 画像。test_operational2）を構築した。



図 87 test_original2 と test_operational2 にある入力データの例

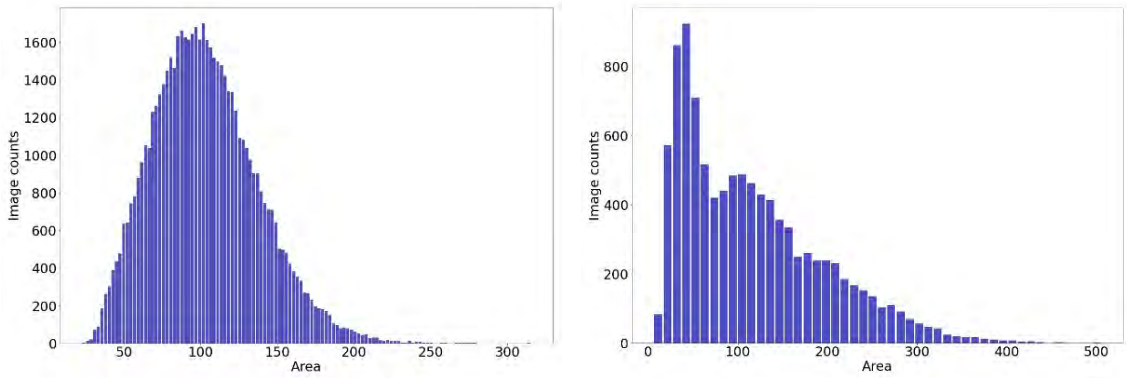


図 88 訓練用セットとテスト用セットそれぞれの特徴量面積に基づくデータ分布

精度(KPI)モニタリング

定期的に、多様なテスト入力(test_original1)を用いて訓練済みモデルを評価する必要がある。テスト用セットは、問題領域内のすべてのクラスとケースの組合せに対するデータを含む必要がある。例として、特徴量**面積**を完全に被覆する均等なクラス分布の test_original1 を示す(図 89)。

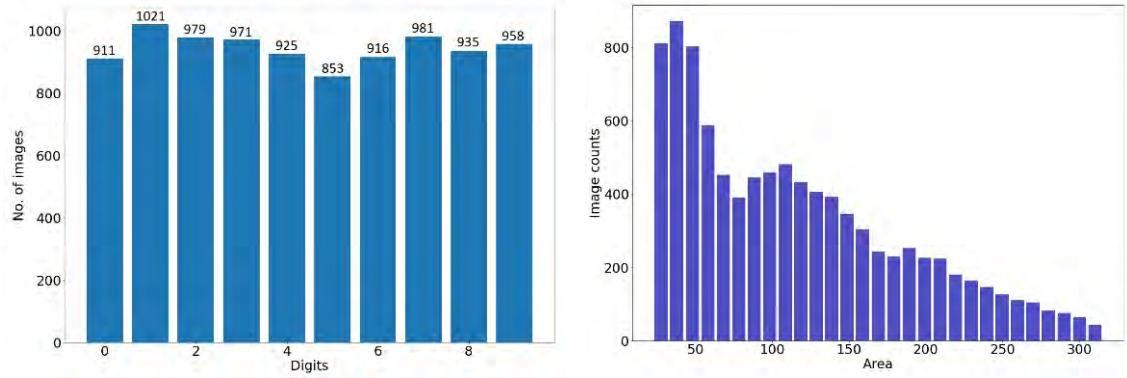


図 89 データセット test_original1 のクラス分布とデータ分布

次に、test_original2 は、数字に関して均等な分布があり、コントラストが高い画像を持つ完全なテスト用データセットである。表 50 は精度モニタリングの結果を示している。

表 50 精度モニタリングの結果

テスト用セット	画像枚数	精度
test_original1	9,450	99.21%
test_original2	10,000	99.20%

継続的なデータ収集とラベリング

運用中、モデルは常に新しい、あるいは未知の入力を取得する。問題空間における現在のデータ分布を分析するために、これらのデータを保存する必要がある。この作業は、データの収集とラベリングからなる。多くの場合、ラベリングは手作業で行われ、高いコストがかかる。この段階では、新しいデータセットが運用中のインプットを基に構築される。例え

ば、テスト用セット一式と test_operational2 が運用入力のセットとなり得る。

モデルへの入力の新規性分析

運用中入力のセットから、被覆率と分布を分析することによって、データの新規性を測定する必要がある。また、距離や類似度に基づく手法で新規性を特定することもでき、それにより、これまでの頑健性指標に基づくモデルのパフォーマンスを推定できるかもしれない。こうして、新規性のあるデータセットを構築することができる。

ここでは、上記で問題領域外と説明したデータセット test_operational1 と test_operational2 を取り上げる。これらをここでのメンテナンスにおける新規データセットの例とする。表 51 から、新規データが学習済みモデルに与える影響が分かる。

表 51 新規データの効果

テスト用セット	画像枚数	精度
test_operational1	550	99.09%
test_operational2	10,000	97.66%

再訓練の必要性を分析する

最高のパフォーマンスを発揮するモデルを訓練しても、時間とともに劣化してしまう。これは入力の変化や環境の変化により発生する。上の表から、test_operational1 はそれほどでもないが、test_operational2 は訓練済モデルにかなりの悪影響を与えていることがわかる。したがって、モデルを再訓練する必要がある。しかし、再訓練を行う際には、モデルが以前学習したことを忘れてしまわないように注意する必要がある。

例えば、訓練用画像（6 万枚）とその低コントラスト版（6 万枚）を用いて同じモデルを再訓練し、test_original2 と test_operational2 で評価した。その結果は以下の表の通りである。

表 52 再訓練の結果

テスト用セット	画像枚数	元の精度	最新精度
test_original2	10,000	99.20%	99.40%
test_operational2	10,000	97.66%	99.40%

モデル出力のモニタリング

モデルの出力は、それが有限かつ有効な結果を与えるかどうかをチェックするために分析する必要がある。間違った予測はもちろんのこと、モデルが特定不能の数値（つまり NaN 値）を出力することもある。そこで、特に再訓練後に、この点についてモデルの検証を行う必要がある。

例えば [47]には、NaN 値をもたらす入力を見つけるためのツールについて記述がある。まず、入力コーパスから画像を選択し、ノイズを加えて変異させる。そして、その画像を ML モデルに通し、出力値と活性化ベクトルを計算する。もし出力値が NaN であればプログラムは停止し、そうでなければ活性化ベクトルを、最近傍アルゴリズムを用いて以前の実行と比較し、新しいカバレッジを決定してその変異した画像を入力コーパスに追加する。次の実

行では、コーパスから最新の要素を抽出する。

追加データセットの作成

AI 問題（郵便番号分析）のソリューションの対応範囲を広げるには、新規または追加のデータセットを作成し、モデルを再訓練することが唯一の方法である。例えば、米国の郵便番号分類器を日本に持ち込んだ場合、同等の性能を得るためには、日本の手書き数字を使って再訓練する必要がある。

おわりに

機械学習技術のメンテナンスは、そのモデルの精度と頑健性の両方を向上させるのに役立つ。また、長く使えるモデルの構築、実世界のデータ分布、コーナーケースの構築にも役立つ。したがって、メンテナンス手続きを励行することで、より良い機械学習ソリューションを徐々に開発することができる。

10 住宅価格分析

10.1 はじめに

本節の目標は、機械学習品質マネジメント（MLQM）ガイドラインに沿った、住宅価格データセットの内部特性の評価方法を示す実装例を作成することである。ここでは、Kaggle データセットにある住宅価格問題、すなわち、住宅価格を予測する回帰問題について説明する。本報告が AI を用いた同様のシステムの評価にガイドラインを適用する際の参考になると期待する。

MLQM ガイドラインを、AI を用いた製品に適用することで、以下のような効果が期待される。

- 住宅価格問題の詳細分析とリアルな事例の掘り下げ
- データ管理および特徴削減のためのさまざまな技術
- 本件の目的にかなう機械学習アルゴリズムの導入
- 製品のエンドユーザーに対する品質、安全性、信頼性の検証

10.2 ビジネス要件の詳細

10.2.1 ユースケース

住宅価格の予測は、人々が住宅を購入する際に、将来の価格帯を示すことで、資金計画を立てるのに役立つと期待される。このモデルへの入力は住宅の特徴である。出力として、このモデルは、住宅の設備にみあう、その住宅の価格を提供する。

10.2.2 背景

住宅価格予測は、不動産開発会社が住宅の販売価格を決定するのに役立ち、顧客が住宅を購入する適切なタイミングを決めるのに役立つ。住宅の価格に影響を与える要因は、立地、物理的条件、建築年などいくつかある。この問題を解決するために、不動産会社は予測誤差を最小限に抑えて住宅の価格を予測する機械学習モデルを求めている。

10.2.3 目的・目標

- ある家の特徴から、その家の価格を推定する。
- 売り手と買い手の双方に住宅価格の推定ツールを提供する。

10.2.4 この製品のステークホルダー

このリファレンスガイドで考慮する、この製品のステークホルダーは以下の通りである。

- 不動産購入者と売却者
- 価格予測サービスを提供する不動産会社

10.2.5 ステークホルダーの初期要求

- 不動産購入者、売却者、そして不動産会社は、この製品が特定の地域の住宅価格を適切に推定することを求める。
- 不動産購入者、売却者、そして不動産会社は、この商品が住宅価格に影響を与える様々な属性を考慮することを期待する。
- 不動産購入者、売却者、そして不動産会社は、製品が使いやすく、理解しやすいものであることを期待する。

10.2.6 ビジネス要件の詳細

開発する製品のビジネス要件は、以下の通りである。

機能要件

最終的な AI システムの機能要件は以下の通りである。

- AI モデルが、家の特徴に関する提供された情報をもとに、家の価格を推定する。

非機能要件

実運用した AI システムの非機能要件を以下に示す。

- AI モデルは、アイオワ州の住宅について良好な性能を発揮する。この地域の外の住宅は対象外である。
- 安価なものから高価なものまで、幅広い価格帯の住宅を考慮する。
- なんらかの特徴量の値が欠けていても、AI モデルは予測を継続する。
- AI モデルは、非常に稀な特徴量を持つ住宅の価格推定に際しても頑健である。

依存事項

- この製品に関する依存事項はない。

制約事項

- データ制約：所有者の個人情報は、物件の価格に影響する可能性があるにも関わらず、特徴リストには含まれていない。

リスクと懸念

- 予測を誤ると、市場での流通を混乱させる恐れがある。

10.2.7 外部品質に関する要求事項

開発する AI ソフトウェアに期待される品質要求レベルを、3 つの主要な外部品質について以下に示す。

安全性

- この製品に関する人身事故の危険はない。
- 新築優良住宅の価格予測が正確でなければ、経済的損失が発生する可能性がある。

パフォーマンス

- 最終的な AI システムは、ステークホルダーと開発者が合意した KPI 指標の閾値を満たす必要がある。
- システム全体として、正確度と精度のバランスが求められる。

公平性

- 製品・サービスの公平性については、識別可能な要件は存在しない。

10.2.8 外部品質特性レベルを定義する

表 53 実現すべき外部品質特性レベル

外部品質	補足説明	想定される深刻度	実現すべきレベル
安全性	人的リスクに対する AI 安全レベル	物理的な被害は想定されない。	AISL 0
	経済的リスクに対する AI 安全レベル	軽微な利益損失、人による監視で回避可能	AISL 0.2
パフォーマンス	一般的 AI 性能レベル	KPI は事前に特定されるが、各 KPI の閾値は他の要因によって変動する可能性があり、ベストエフォートで提供される	AIPL 1
公平性	一般的 AI 公平性レベル	製品・サービスに対する明確な要件はない	AIFL 0

10.2.9 おわりに

本節では、ビジネスの観点から、このソリューションの背景にある目的を説明した。またその一環として、ユーザーのニーズと期待、および実運用の成否に関わる高レベルの制約を述べた。また、住宅価格を予測する AI 利用製品に対するビジネス上の要求も明示した。これは、開発者が次節以降で内部品質を評価するのに役立つ。

10.3 製品仕様

製品仕様として提案しうる内容は、以下のように記載できる。

10.3.1 モデル仕様

- 学習の種類：教師あり学習
- AI モデルの種類：回帰
- モデルのアーキテクチャ：シンプルな深層学習
- 実行するタスク：価格予測

10.3.2 データ関連仕様

- データに関する仕様：考慮すべき属性／無視すべき属性／あるいくつかの特徴に対する指定値、リスト

10.3.3 KPI 仕様

- 精度：LRMSE, MSE, RMSE など。

10.4 データセットの紹介

10.4.1 データセットの探索

ここでは、住宅販売価格を予測する回帰問題のために House price データセットを選択した。この問題の分析には Kaggle [48]のデータセットを利用する。また、必要に応じて独自のデータセットを構築することも可能である。

10.4.2 入力データのサンプル

さて、 GarageQual (ガレージ品質)は、モデルへの単純な入力データである。ガレージ品質は、住宅価格分析のための特徴の一つである。データセットにあるこの属性の値は以下の通りである。データセットの説明では、GarageQual のデータポイントの総数は 1379 個、値は以下に挙げる 5 通りである。Gd (Good, 良い), TA (Typical/Average, 典型的または平均的), Fa (Fair, まあまあ), Po (Poor, 悪い), NA (No Garage, ガレージなし). 典型的または平均的なデータの数は 1311 と多く、その他はわずかである。

特徴名：GarageQual

dtype: object (ある属性の記述)

- top 1379 (データポイントの総数)
- unique 5 (異なる値の個数)
- top TA (最頻値)
- freq 1311 (最頻値の個数)

10.5 MLQM ガイドラインを用いた品質保証手順

今回のデータセットを簡単に調べた後、次は、MLQM ガイドラインで言及されているリスク回避と AI パフォーマンスという 2 つの外部品質の達成度に対して、品質マネジメントの 9 つの特性軸（内部品質）をそれぞれ探っていくことにする。

- A-1: 問題領域分析の十分性
- A-2: データ設計の十分性
- B-1: データセットの被覆性
- B-2: データセットの均一性
- B-3: データの妥当性
- C-1: 機械学習モデルの正確性
- C-2: 機械学習モデルの安定性
- D-1: プログラムの信頼性
- E-1: 運用時品質の維持性

10.5.1 A-1：問題領域分析の十分性

定義

問題領域の十分性分析では、従来のソフトウェアにおけるリスク要因の分析や、ブラックボックステスト実施時にそれらのリスク要因を含めるためのテスト要求分析を行う。システムが対応すべき様々な状況に対して十分な訓練用データとテスト用データを確保する

ために、「データ設計の十分性」としてデータ設計を隅々まで検討する必要がある。具体的には、訓練データの作成からテスト工程までの段階で取り扱う、属性値の組合せの数と内容をこの段階で検討する。

問題領域分析の十分性に関する分析の一般的なプロセスまたは構造は以下の通り。

- 問題領域を定義し、すべての範囲のデータがあるかどうかを確認する
- 問題領域におけるコーナーケースを特定する
- 様々な特徴選択法を適用して重要な特徴を選択する
- 選択した特徴について許容する範囲を設定する

問題領域の定義

住宅価格の問題には、79 の特徴と 1460 のデータポイントがある。まず、これが問題領域であって、次元が 79 ある、ということになる。

ありうるすべての価格帯のデータ

ありうるすべての価格帯のデータがあるかどうかを確認する必要がある。例えば、2020 年の米国国勢調査 [49]によれば、アイオワ州の居住されている住宅 906,967 軒中、$1,000,000 以上は 0.6%しかないので、データセットになくても影響は小さいと考えられる。一方、$50,000 未満の住宅は 8.4%もあり、この範囲のデータはデータセットに含めるべきだと思われる。

適切な特徴量次元を選択する

属性とそれに対応する属性値は、データ固有のありうるシナリオのうち、考慮すべき、また後の分析（被覆性や十分性など）のために列挙すべきものをカバーする必要がある。

ここでは特徴が 79 個あり、これを分析して除外する特徴を決定し、新たに追加するべき特徴があるかどうかを確認する必要がある。特徴量の削減には、PCA、相関行列、変数削減法など、さまざまな方法を適用することができる。また、新しい特徴を追加するためには、多くの労力が必要である。

例

Kaggle: House Prices のような問題では、多くの特徴があり、必要なものも冗長なものもある。この特徴空間を減らせば、MLQM ガイドラインに沿ったデータセットの品質評価は楽になる。そこで、元のデータセットに対する次元削減として特徴選択のみを行う。不要な特徴を削除するだけだが、これにより説明可能な属性からなる、より小さな特徴空間が得られる。

相関行列によるフィルタリング

フィルタリングのために、まず、数値データを抽出した。次に、数値データセットの相関行列を計算した。

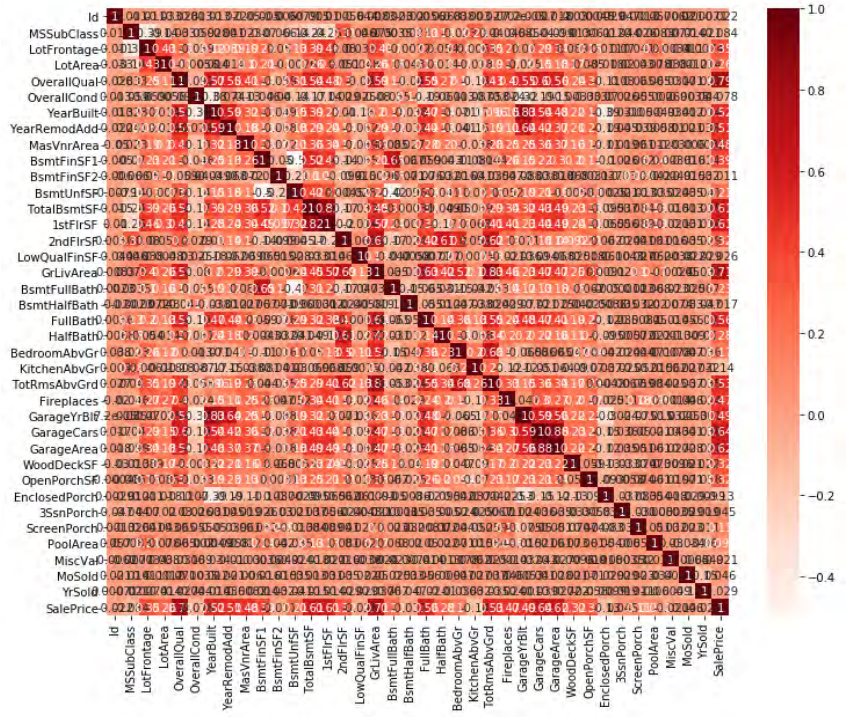


図 90 元の数値データセット(38 X 38)の相関行列

ここでは、38 個の数値属性があり、その中から SalePrice (販売価格)と 0.5 以上の相関係数を持つ属性を選択する。その結果、11 の属性が得られ、縮小した相関行列は以下のようになる。

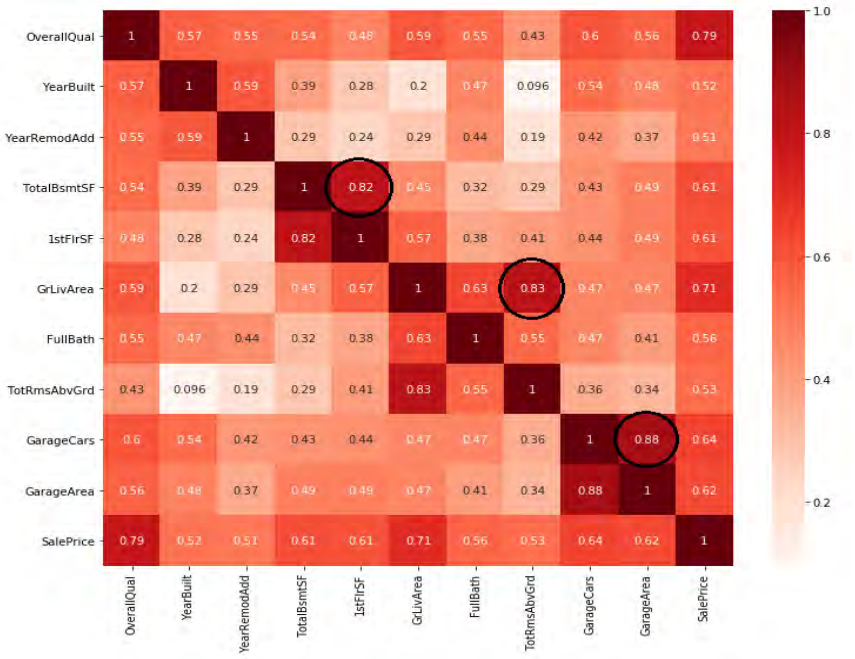


図 91 選択した特徴の相関行列（11 X 11）

これは、元の行列よりもずっとシンプルである。それでも、 相互の相関が高い（>0.8）属性の対がいくつかある。そこで、これらの対について、SalePrice との相関がより高い片方だけを残した。その結果、属性は SalePrice を含めて 8 個に絞られた。選択した数値属性は以下の通りである。

OverallQual', 'YearBuilt', 'YearRemodAdd',
'TotalBsmtSF', 'GrLivArea', 'FullBath',
'GarageCars', 'SalePrice'.

このようにして、数値属性の数は 36 から 7 に減少した。カテゴリー属性についても同様の削減が可能である。

この 7 つの属性はすべて分析する価値があるが、ここでの分析にはまだ多すぎる。そこで、以下の 2 つの特徴だけを、明確に定義された特徴次元として使用する。

'GrLivArea' この属性は、平方フィートでの全地上生活面積である。これは、家に関する非常に一般的な情報であり、重要なものである。

'ExterQual' 前の属性と同様に、今回使うデータセットにある、この非数値的属性のカテゴリーを列挙した。

範囲内・範囲外の選択

問題領域で考慮すべき選択した特徴の値の許容範囲を具体的に宣言する必要がある。ここでは、ユーザーの要求を優先する必要がある。設計者として、どのような値を対象範囲内とし、どのような値を除外するかを定義する。例えば、特徴 GrLivArea は、販売価格と関係が深い。2019 年の米国国勢調査 [50]によれば、米国全土で最も件数が多いのは 1,000-1,499 平方フィートの区間で、1 人当たりの面積の中央値は 700 平方フィートである。アイ

オワ州の住宅の平均面積は 1,550 平方フィートである [51]。州都デモインでは、新しいゾーニング法により、1,100 平方フィート以下の小さな家は建てにくくなり、1800 平方フィート以上の一戸建ては建てやすくなっている。しかし、ここで扱うデータには 1930 年頃の古い、小さな家が入っている。そこで、地上生活面積の受け入れ範囲を 300-5000 平方フィートとした。

また、ExterQual は非数値的な特徴で、属性値として以下の値を取る。Ex = Excellent (とても良い), Gd = Good (良い), TA = Typical/Average (典型的・平均的), Fa = Fair (まあまあ). ソリューションデザイナーの判断として、これをそのまま許容範囲とする。

不健全ケースの特定

不可能と思われる属性の組合せは、分析から除外する必要がある。例えば、プール付きの家であっても、プール付きの家として十分な面積がない場合がある。

おわりに

本節では、79 個の特徴量から分析に用いる特徴量 2 つ、GrLivArea と ExterQual を選択し、この 2 つの特徴量の説明を行った。これで問題領域の分析が完了し、その範囲を、必要な領域をすべて含むように定義した。実在するデータはすべてこの特徴量空間に収まるはずである。これで、この要件は満たすことができた。

10.5.2 A-2: データ設計の十分性

「データ設計の十分性」の定義。

ここでいう**データ設計の十分性**とは、機械学習ベースのシステムが実世界で使用される場面について十分な要求分析が行われ、その分析結果が起こりうるすべての状況を網羅していることを意味する。

データ設計の十分性のための一般的な分析プロセスまたは構造は以下の通り。

- 定義した問題に対するデータセットを選択する
- 必要に応じて、各種の拡張・アノテーションルールを適用して、新しい特徴を追加する
- 想定する範囲に十分なデータがあるかどうかを確認する。

各特徴次元についてのデータ管理

上記で定義した問題領域から、属性値の可能な組合せの総数を算出する必要がある。

- ここでは、問題領域に適合するようにデータセットを設計する必要がある。独自にデータセットを作成するのは非常に難しいので、利用可能なデータの中から問題領域に合うデータセットを選択する。ここでは、Kaggle の住宅価格分析データを選択する。まず、そのデータセットが問題領域と似ているかどうかを確認する。類似性のチェックについては、**データセットの被覆性**の節で被覆性や分布をチェックする。
- 選んだデータセットが期待通りならよいが、利用可能なデータが十分でなければ、

何らかの拡張をする必要がある。既存データセットに、既存の属性とは関係のない新しい特徴を加えたいなら、アノテーション等の作業を人手で行う必要がある。

- 住宅価格は不連続なデータセットなので、数値的な拡張は実のところ不可能である。
- あり得るすべての価格帯のデータが揃っているかどうかを確認するために、販売価格の分布を確認する必要がある。ここでの推定対象はアイオワ州の住宅価格で、以下に示すこのデータセットの販売価格の分布も、アイオワ州に関するものである。

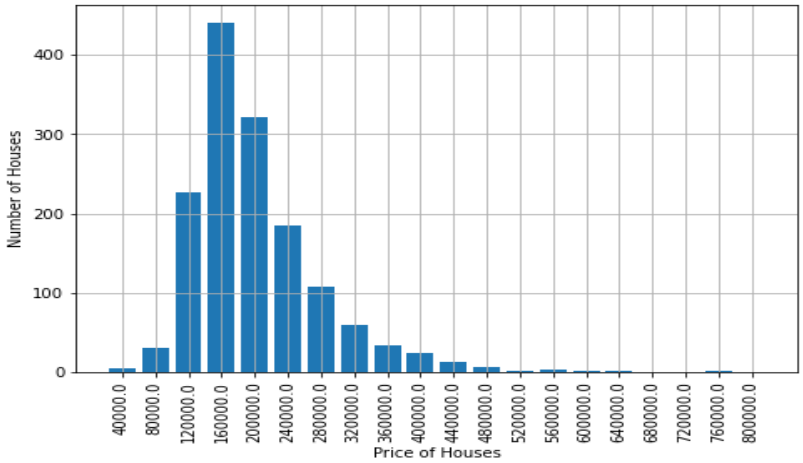


図 92 販売価格の分布

データセットを設計する際には、膨大な数の特徴の組合せが見つかる。ソリューション設計者は、重要な組合せとそうでない組合せを見分ける必要がある。問題領域の分析に基づいて、重要度の低い特徴を組合せて有用な特徴一つにしたい場合には、数値的な手法を使う。これにより、特徴空間の次元と複雑さを削減でき、ソリューション設計者は、重要度の低い、あるいは不適切な、またはリスクの高い組合せを排除することができる。

おわりに

Kaggle House Prices は回帰問題であり、その特徴空間は広大である。新しい素性を追加することも、データ拡張も、特徴削除も不可能である。データ設計するには、このデータは手ごわい。独自のデータセットを定義したり、作ったりすることができれば、問題は解決する。しかし、そのようなことをする人手も時間もないので、このデータセットを分析に使うことにする。

10.5.3 B-1: データセットの被覆性

データセットの被覆性とは、前節で述べた、基準とする分布を設定することにより想定範囲に入れたケースに対して、十分な量のデータが与えられており、それらのケースに対応する、ありうる入力に対して、見落としがないことをいう。

データセットの被覆性に関する一般的な分析のプロセスまたは構造は以下の通り。

- 選んだ組合せのカバレッジを確認する

- レアケース、コーナーケースを特定する
- 問題領域から排除できる特徴や値の範囲があるかどうか調べる。

各組合せのカバレッジ

検討する特徴の組合せごとに、データカバレッジを計算し、あらかじめ設定したカバレッジ基準と照合する必要がある。これにより、訓練用データセットの範囲とテスト用データセットの範囲が決まる。

- 定義した問題領域におけるすべての特徴次元の範囲の組合せによってカバーされる特徴空間におけるデータセットの分布を見る必要がある。データセットが各組合せをどの程度カバーしているかを確認する必要がある。
- 2 つの特徴次元にわたる訓練用データの完全な分布を以下の表に示す。

表 54 訓練用データの特徴空間上の分布

	ExterQual			
GrLivArea	Ex	Gd	TA	Fa
5001-6000	1	0	0	0
4001-5000	2	1	0	0
3001-4000	2	5	6	1
2001-3000	25	103	68	0
1001-2000	21	359	630	5
0-1000	1	20	202	8

- GrLivArea の定義範囲は、300 から 5000 平方フィートである。この表から、この特徴の値の全範囲についてデータがあると分かる。これは、データがこの次元を完全に被覆していることを意味する。

レアケース・コーナーケースの見極め

ある組合せが、それに対するデータポイントが足りないが、極めて重要なケースであることがある。これらはレアケースとかコーナーケースと呼ばれる。このようなケースについてはデータを生成または収集する必要を判断する必要がある。

- 各領域が何らかのデータでカバーされているかどうかを確認する必要がある。例えば、GrLivArea が 5000 から 6000 までの領域には、データポイントが 1 つしかないため、レアケースまたはコーナーケースとみなせる。

特徴削除

場合によっては、データポイントが希少であっても、システム全体にはほとんど影響を与えないこともある。そのような稀なケースを識別するには、モデルの評価段階において訓練から完全に除外しつつテストからは除外しないでおいてみる。もし結果が同じであれば、その特徴は不要と言える。

- 表から分かる通り、GrLivArea の 5000 から 6000 の領域には、データポイントが 1

つしかなく、この領域はあまりカバーされていない。この範囲を削除したり、小さくしたりすることはできるが、そうするとモデルに制限が生じる。

おわりに

この問題では、2 つの特徴を選んだ。これらは、いくつかの領域では十分にカバーされておらず、それらの領域ではモデルがうまく機能しないだろう。それらの領域にデータを追加できるとよいが、今回それはできないので、このデータセットはその領域のモデルの訓練やテストには適していない。つまり、今回選んだ問題に対し、データセットの被覆性は十分でないと言わざるを得ない。このデータセットはこの内部品質検査に関して不合格である。

10.5.4 B-2: データセットの均一性

定義

先に述べた**被覆性**と対峙する概念として、想定される入力データ全体に対するデータの**均一性**がある。データセット内の各状況や事例が、入力データ全体における出現頻度に従って抽出されている場合、データが「均一」であるとみなす。

データセットの均一性のための一般的な分析プロセスまたは構造は以下の通り。

- 選択した特徴の分布を確認する
- 予想される分布と実際の分布を比較してみる
- 分布を見て、必要であれば問題領域を更新する
- 最後に、判断する。

各ケースに十分なデータがあるか

想定しうるあらゆるケースのそれぞれに十分なデータがあることを確認する必要がある。データポイントの分布を測定し、予想される分布と比較して分析すること。データポイントの分布が実世界の分布に従っていることが望ましい。

例　訓練用データセットの、選択した属性の対象領域における、望ましい均一な分布を図 93 に示す。

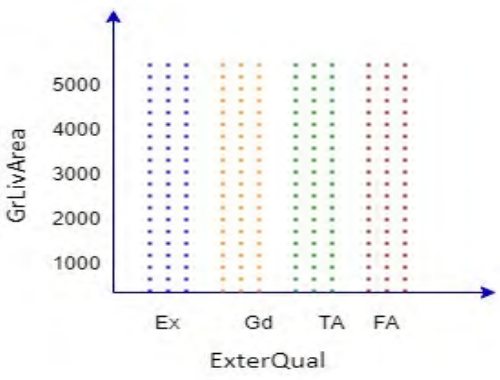


図 93 選択した特徴空間における住宅の望ましい分布

実際の分布を図 94 に示す。

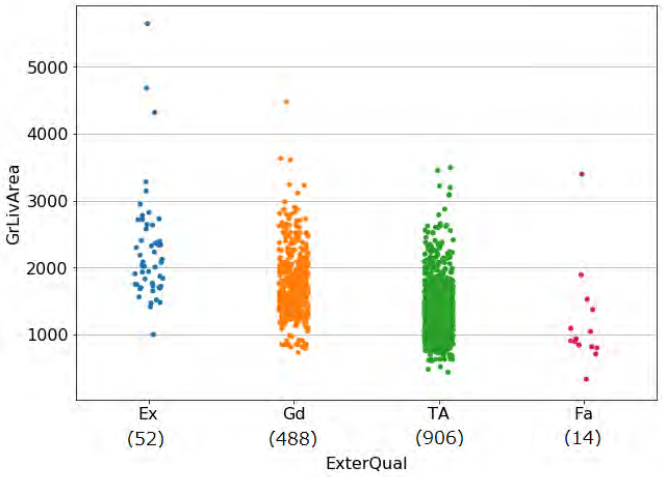


図 94 選択した特徴空間における訓練用データの実際の分布

この分布図から、1000 平方フィート以下の Ex 住宅はなく、4000 平方フィートから 5000 平方フィートまでの住宅は非常に少ないことがわかる。同様に、この範囲の Gd 住宅はさらに少なく、TA 住宅はこの範囲にはない。

外観品質（ExterQual）が Fair の住宅はわずかしかなく、定義した問題領域の中では比較的小さい住宅である。また、住宅の地上生活面積(GrLiveArea)は、ほとんどが 1000～3000 平方フィートである。

おわりに

ここでは、期待される分布と実際の分布が一致していない。つまり、分布は一様ではないと言わざるを得ない。問題領域はよく被覆されているが、データポイントは一様には分布していないと言える。ここで選んだ問題は、この品質検査測定には不合格である。

10.5.5 B-3: データの妥当性

この版のリファレンスガイドでは本品質を検討対象としていない。

10.5.6 C-1: 機械学習モデルの正確性

機械学習モデルの正確性の定義

機械学習モデルの正確性とは、学習データセット（訓練用データ、テスト用データ、バリデーション用データからなる）に含まれる特定の入力データに対して、機械学習要素が期待通りに反応することを意味する。

機械学習モデルの正確性に関する分析の一般的なプロセスまたは構造は以下の通り。

- パフォーマンスチェックのための KPI を選択する
- KPI でモデルの性能を確認する

- モデルの性能が十分でない事例を探す
- すべての分析を行った上で判断する

特定の方式を選択する

評価方法は、訓練用データに対する収束性とテスト用データに対する達成度の両方について記述すること。

例　まず、パフォーマンスチェックのための KPI を選択する必要がある。評価用の KPI には様々な種類がある。

- 平均二乗誤差（Mean Square Error）
- 二乗平均平方根誤差（Root Mean Square Error）
- 平均絶対対数誤差（Mean Absolute Logarithmic Error）

平均二乗誤差　統計学では、平均二乗誤差（MSE）は、実際の値と推定値の差の二乗の平均と定義される。これは、回帰モデルのモデル評価指標として用いられ、値が小さいほど適合度が高いことを示す。平均二乗誤差を用いると、誤差は 3.211 となった。

$$Mean\ Square\ Error = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{33}$$

二乗平均平方根誤差　二乗平均平方根誤差(RMSE)は残差(予測誤差)の標準偏差である。残差は、データポイントが回帰直線からどれだけ離れているかの尺度であり、RMSE は、これらの残差がどれだけ広がっているかの尺度である。言い換えれば、データがベストフィットの線の周りにどれだけ集中しているかを示している。二乗平均平方根誤差は、気候学、予測、実験結果の検証のための回帰分析などでよく使われている。RMSE を用いると、誤差は 1.791 となった。

$$\boldsymbol{Root\ mean\ Squared\ Error} = \sqrt{\sum_{i=1}^{n}\frac{(\hat{y}_i - y_i)^2}{n}} \tag{34}$$

平均絶対対数誤差　測定値と「真の」値との差である。販売価格の分布は散らばっているので、対数誤差を使用する。平均絶対対数誤差を用いると、誤差は 0.7653 となった。

$$\boldsymbol{Mean\ Absolute\ Logarithmic\ Error} = \frac{1}{n}\sum_{i=1}^{n}|log\ y_i - log\hat{y}_i| \tag{35}$$

以上の分析で、異なる種類の KPI に基づくさまざまな測定方法を見てきた。次に、**異なるサイズのデータセットで訓練したモデルの性能を、**平均絶対対数誤差を用いて比較する。まず、データセットカバレッジテーブルの異なる領域のデータを用いて、全結合の dense **層**の訓練と評価を行った。その結果を下表に示す。

表 55 平均絶対対数誤差

	ExterQual							
GrLivArea	Ex		Gd		TA		Fa	
	データ	誤差	データ	誤差	データ	誤差	データ	誤差

5001-6000	1	-	0	-	0	-	0	-
4001-5000	2	-	1	-	0	-	0	-
3001-4000	2	-	5	134.567	6	8.69171	1	-
2001-3000	25	8.95012	103	6.51191	68	18.5707	0	-
1001-2000	21	40.3462	359	1.44147	630	1.33367	5	46.6369
0-1000	1	-	20	4.42439	202	3.72669	8	41.8109

ここで、誤差は利用可能なデータを用いた分割数 4 の交差検証から得られた平均絶対対数誤差である。訓練用データ数に対して誤差の結果をプロットしたものを示す。

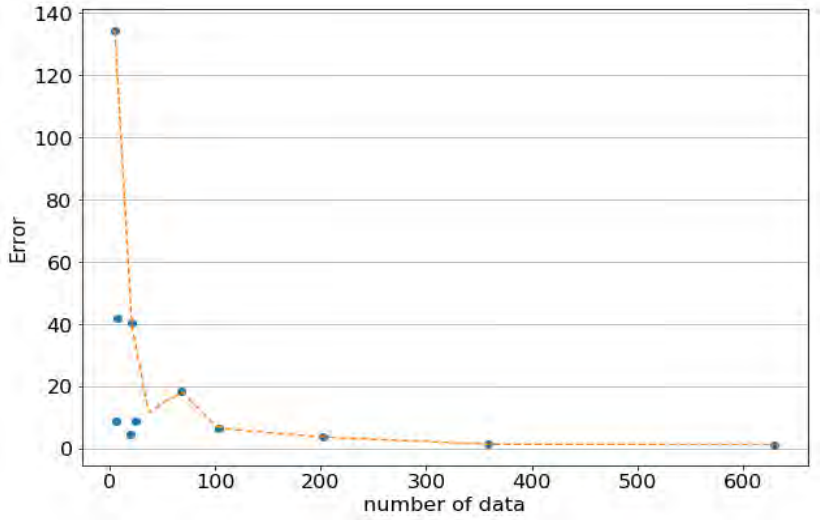


図 95 誤差と訓練用データ数の補間グラフ

指数関数的に減衰していくようなカーブで、予想通りである。誤差はデータ量に反比例している。

$$\textbf{Error} \propto \frac{\mathbf{1}}{\boldsymbol{number\ of\ data}} \tag{36}$$

データ分布の違いに関する性能比較

上記の場合、カテゴリー属性は訓練時に使われておらず、役に立っていなかった。ここで、属性の分布の違いがモデルの性能にどのような影響を与えるかを見てみたい。この分析のために、我々は以下のようなデータのサブグループを作成した。

表 56 GrLivArea に沿ったデータの分布。

	ExterQual			
GrLivArea	Ex	Gd	TA	Fa
5001-6000	1			
4001-5000	3			
3001-4000	14			

2001-3000	196
1001-2000	1015
0-1000	231

ここで、1,460 点のデータを全て使うこともできるが、その場合、GrLivArea の分布もモデル性能に影響を与えることになる。図 96 に示すこの属性の分布図から分かる通り、ほとんどのデータは 1001-2000 平方フィートの間にある。この範囲に限定することで、GrLivArea に関するデータサンプルの分布の訓練や評価への影響を最小限に抑えることができる。

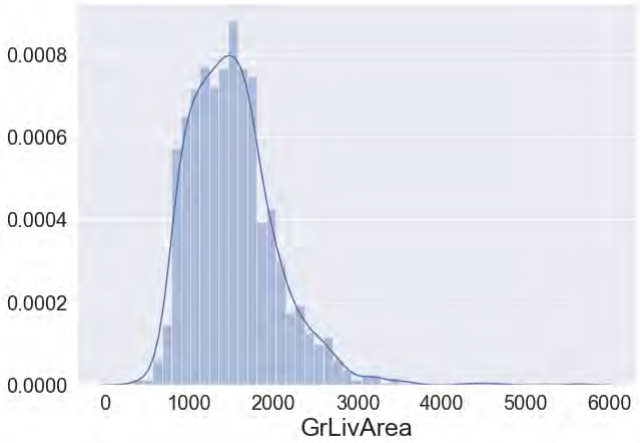


図 96 GrLivArea を基準としたデータサンプルの分布

そこで、最もデータサンプル数の多い *1001-2000 sq ft GrLivArea* をサブグループとした。選択したサブグループの属性 ExterQual の分布を図 97 に示す。

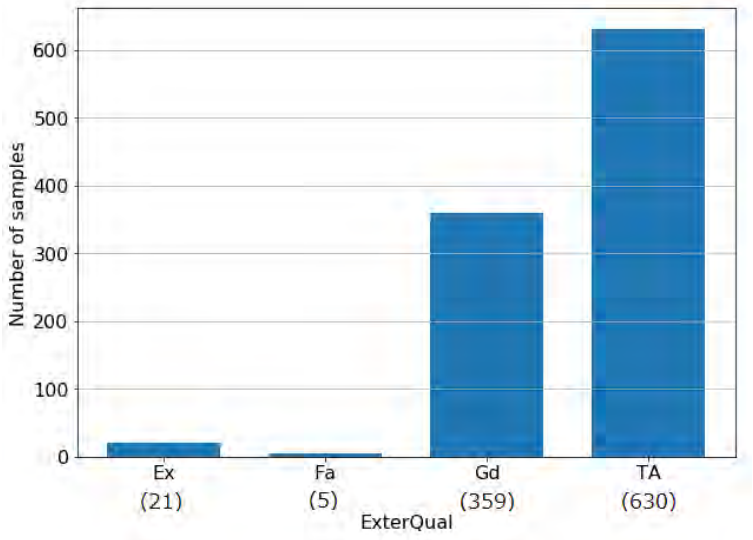


図 97 訓練用データのサブグループ（GrLivArea: 1001-2000 sq ft）での ExterQual の分布

データ分布の影響を把握するため、カテゴリーの組合せ 3 種類について、前回と同じモ

デルの訓練と評価を行った。得られた結果と用いた組合せを表 57 に示す。

表 57 異なるカテゴリーの組合せのデータで訓練した結果

カテゴリーの組合せ	最も稀なデータの件数	誤差
TA + Fa (630 + 5)	5	1.28569
TA + Ex (630 + 21)	21	1.18049
TA + Gd (630 + 359)	359	0.91479

ここで、「最も稀なデータ数」とは、ある組合せで最も稀なカテゴリーに存在するデータの数である。

これを見ると、最も稀なデータの数が少ないと誤差が大きく、逆に最も稀なデータの数が多いと誤差が小さいことが分かる。つまり、データセットにレアケースを含めることは必要だが、レアケースのデータサンプルは十分な量必要である。

ここまでは、データ分布を見てコーナーケースを測定してきた。以下ではモデルの性能を測定することで、コーナーケースを特定することができる。

おわりに

予測精度は、KPI で示すことができる。ここでは、KPI の測定に平均絶対対数誤差を使用した。今回のモデルは、いくつかの領域が十分被覆されていないため、うまく機能しないことがある。上記の分析から、データカバレッジが高いほど、誤差は少ないと言える。また、データが均一なほど、パフォーマンスが高いとも言える。

10.5.7 C-2: 機械学習モデルの安定性

定義

機械学習モデルの安定性とは、機械学習要素が、学習データセットに含まれない入力データに対しても、学習データセットに含まれるデータと十分似た反応を示すことを意味する。機械学習要素の振る舞いの予測可能性は、汎化能力の低さや敵対的データによる予測不可能な振る舞いを除去することで改善する。安定性は機械学習のライフサイクルと強く関連しているため、安定性の目標を達成するためには、主に以下のフェーズで評価・強化する必要がある。

機械学習モデルの安定性のための一般的な分析プロセスまたは構造は以下の通り。

- 性能チェックのために別のモデルを選択する
- パラメータチューニングを行い、ロスカーブを確認する
- 最後に、判断する

例

訓練用データセットとバリデーション用データセットを分離することによって訓練用データセットの過学習を避けるためにいくつかの反復訓練フェーズを設ける。入力の最小限の変化が出力に与える影響を評価した後、訓練プロセス全体を監視する。さて、我々は「住宅価格分析」問題に対して、全結合モデルを選択した。モデルの構成を表 58 に示す。

表 58 住宅価格分析に使用したモデルのアーキテクチャ

アーキテクチャ	FC(128)+ReLU FC(256)+ReLU FC(64)+ReLU FC(10)+ReLU
訓練可能なパラメータ数	88,449

バッチサイズ 32、エポック 300 において、訓練ロスは 0.0212、バリデーションロスは 0.0514 となった。図 98 に訓練とテストのロスカーブを示す。

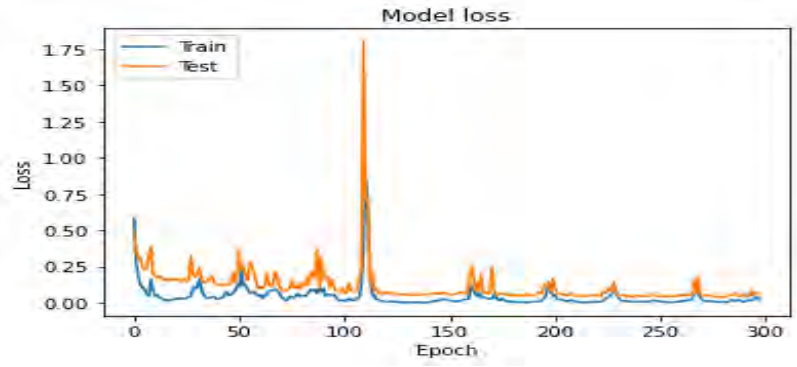


図 98 単純な訓練とテストのロスカーブ

図 99 では、同じモデルアーキテクチャでもハイパーパラメータを調整することで、エポック 100 において訓練ロス 0.0451、バリデーションロス 0.1237 が得られた。ロスカーブから、訓練カーブとテストカーブが互いに関連していることがわかる。

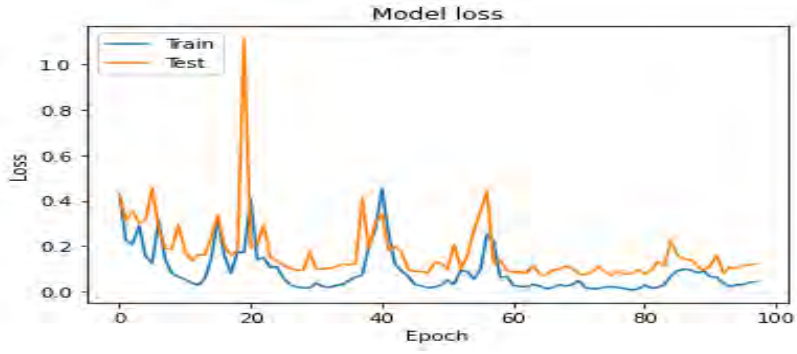


図 99 ハイパーパラメータチューニングによる訓練とテストのロスカーブ

モデルアーキテクチャを変更した後、出力に何らかの影響があるかどうかを確認する。ここで用いたモデルアーキテクチャを表 59 に示す。

表 59 モデルのアーキテクチャ

レイヤー（種類）	出力形状	パラメータ
dense_84 (Dense)	(None, 19)	5776
dense_85 (Dense)	(None, 19)	380
dense_86 (Dense)	(None, 19)	380
dense_87 (Dense)	(None, 19)	380
dense_88 (Dense)	(None, 1)	20

訓練可能なパラメータ数　6, 936

訓練ロス 0.0241　バリデーションロス 0.0494　エポック 100

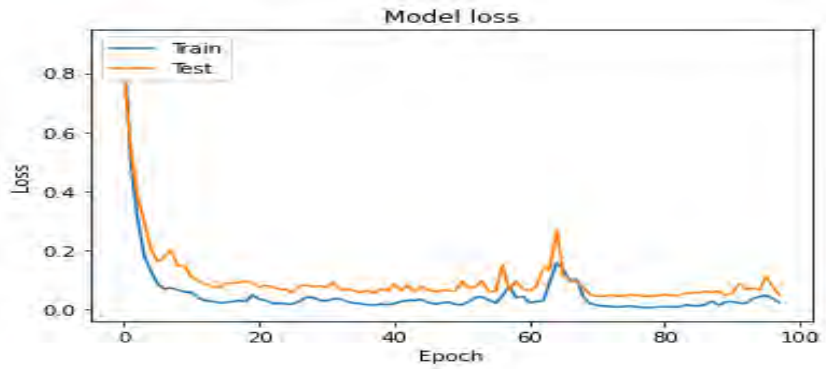


図 100 別のモデルで訓練とテストを行った場合のロスカーブ

これらの出力結果から、モデルのパラメータやアーキテクチャの違いは、出力結果に違いを与えないことがわかる。また、訓練ロスとバリデーションロスは互いに相関しており、過学習の問題は起きていないことが分かる。

安定性の測定に関する研究論文のリンク [52] [53]を挙げておく。

おわりに

内部品質**機械学習モデルの安定性**は、モデルの汎化能力を測定し、コーナーケースやレアケースに対するモデルの反応を評価し、敵対的データに対するモデルの性能を評価することによって改善できる。この品質は、頑健性、つまり未知の入力や新しい環境条件下でのモデルの性能に関するものである。

10.5.8 D-1: プログラムの信頼性

定義

プログラムの信頼性とは、機械学習の訓練段階で用いる訓練ソフトウェア部品や、実行時に用いる予測・推論ソフトウェア部品が、与えられた訓練用データあるいは訓練済み機械学習モデルに対して正しく動作することを意味する。これには、アルゴリズムとしての正しさに加え、メモリ資源制約や時間制約の充足、ソフトウェアセキュリティなど、ソフトウェ

アに求められる一般的な品質要件も含む。

プログラムの信頼性のための一般的な分析プロセスまたは構造は以下の通り。

- 言語
- フレームワーク
- メモリ使用量
- メタパラメータ
- ハードウェア
- ソフトウェアのセキュリティ

言語

ML を扱う際には Python, R, Java, Julia, Scala などがよく使われる。ここでの AI の開発には Python 言語を使用した。Python は様々なオープンソースパッケージを使用しており、それらは互いにバージョン互換である必要がある。そのため、開発者は、使用するパッケージとそのバージョンの一覧を提供する必要がある。

表 60 パッケージとそのバージョンの一覧

プログラミング言語	バージョン
Python	3.6.12
パッケージ	バージョン
NumPy	1.18.5
TensorFlow	2.3.1
Pandas	1.1.5
Matplotlib	3.3.2

フレームワーク

ML モデルは様々なフレームワークで実行することができ、それぞれ実行速度が異なる。最もポピュラーなのは、Keras, TensorFlow, Caffe, Theano, Microsoft CNTK, PyTorch, scikit-learn である。各フレームワークにはそれぞれ特徴があり、異なるニーズに合わせて作られている。TensorFlow, Keras, Theano はニューラルネットワークを非常に高速に実行し、AWS は一般的に堅牢で、scikit-learn は表形式データに最適である。中には、お金を出せばより速い結果を得られるものもある。このプロジェクトでは、TensorFlow と Keras を使用する。

メモリ使用量

ここでは、モデルが使用するメモリ量と、使用可能なメモリ量を表示する。したがって、AI デバイスが動作しているときのメモリの最小使用量と最大使用量を、その際のデータストレージ、モデルのパラメータ、コードやアルゴリズムなどの組合せと合わせて明示する必要がある。

- **モデルアーキテクチャと重み** ここでは Keras で使われることのある HDF（Hierarchical Data Format）ファイル（.h5）を使用して、訓練済み AI ネットワ

ークとその重みを保存する。保存したネットワークには 88,449 個のパラメータがあり、ハードディスクに約 2MB の容量を必要とする。

- **入力データ**　入力とは一般に計算機に何かを提供したり与えたりすることを言う。つまり、計算機や機器が外部からコマンドや信号を受け取ることを、その機器への入力と呼ぶ。したがって、入力データによるメモリ使用量は、マシンの完全な設計後に定義することができる。
- **コードやアルゴリズム**　機械学習における**アルゴリズム**とは、機械学習**モデル**を作成するためにデータについて実行する手順である。機械学習アルゴリズムは、データから**学習**したり、データセットに**当てはめ**たりする。たくさんの機械学習アルゴリズムがある。様々なプログラミング言語で書かれた様々なアルゴリズムが、装置のワークフローの中で使われる。こういうコードは、ハードディスクに大して場所を取らない。
- **再訓練のためのデータセット**　再訓練のためのデータセットを保持するために、少なくとも実際のデータセットと同じ大きさの場所を確保する必要がある。例えば、住宅価格のデータセットは約 449.88KB の容量を必要とする。

メタパラメータ

メタパラメータとは、ML アルゴリズムに入力して、ML アルゴリズムの振る舞いを指示する値である（したがってアルゴリズムの訓練や予測にかかる時間に影響する）。モデルによってメタパラメータは異なる。

- **学習率(η, eta)**　学習率が高いほど、モデルの計算時間は短くなる。
- **仕様の大きさ**　モデルの仕様（NN ではレイヤー数、KNN では k の値など）が大きくなるにつれ、モデルの計算時間も長くなる。
- **ラウンド数／エポック数**　機械学習モデルのラウンド数やエポック数を増やすと、訓練時間が長くなる（しかし予測時間は変わらない）。
- **目的**　ML モデルの中には様々な目的に合わせて調整できるものがある。目的によって学習時間や予測時間は異なる（通常、回帰はカウントよりバイナリが長くかかる）.
- **早期停止**　ML 実装の中には、モデルがバリデーション用データセットで十分な性能を発揮した場合、モデルの訓練を早期に（自動的に）停止することができるものがある。この早期停止機能を追加して実行時間に悪影響はない。
- **その他**　ML モデルはそれぞれ異なるメタパラメータを持ち、それが学習時間に影響するため、注意を要する。

ハードウェア

モデルが動作するハードウェアを変更することは、モデルの動作を速くするための高価だが簡単な方法である。処理装置は主に 3 つある。CPU、GPU、そして TPU である。

TPU（Tensor Processing Units）は、Google Cloud を通じてアクセスできる Google の

専有資産で、常に改良が加えられている。ニューラルネットワークを高速に実行できる。DeepMind の処理ユニットとして使われている。TPU は、既存のどの GPU よりもジュールあたりの入出力演算数が高い。

GPU は CPU より高速である。モデルが動作するプロセッシングユニットの数を増やせば、モデルはより速く学習・予測できるようになる。画像認識アルゴリズムの多くは、GPU 上で特に高速に動作する。画像生成用の GAN の中には、GPU でしか動かないものもある。CUDA の行列計算では、ビット長が異なる場合がある。

例 [10] TensorFlow や Chainer のバージョンによっては、行列計算の精度や計算のためのメモリやリソースの消費を考えると、32bit で十分であり、64bit では多すぎる場合がある。Quadro (nvidia)のバージョンによっては 32bit をサポートしていないものがある。その場合、高価な Quadro GPU で 32bit の計算を高速化しても効果はない。当時は GeForce で 32bit に対応していたので、Quadro の代わりに GeForce を使う人が多かったようである。一方、Intel が AI や ML に特化した命令セットとドライバソフトウェアを公開した。そのため、フレームワーク（TensorFlow や Pytorch など）の提供者は、両方の実装（CUDA 32bit と Intel ドライバー）をサポートする傾向にある。ビジネスソリューションでは 16bit で十分な場合もあり、そもそも GPU を使っていないソリューションもある。今回の住宅価格問題では、GPU を使うなど環境を変えて問題の健全性を確認し、その結果を再現することで両者の違いを比較することができる。今回の住宅価格データセットを用いたモデルの訓練は、2GB の RAM と GPU なしで、まずまずの時間で実行できる。H/W や S/W の最適な構成は、技術進歩によって頻繁に変わるので、設計者は現在の技術情報を検索して、バランスの良い最適な構成を決定する必要がある。

ソフトウェアのセキュリティ

装置がオンラインで動作する場合、セキュリティは重要な問題である。AI・機械学習機能を持つアプリケーションセットを構築する際に、ソリューション設計者が考慮すべき事項については、ガイドラインのセキュリティに関する章（第 2 版では 9 章）を参照されたい。

おわりに

AI ソリューションというからには、コード、アルゴリズム、データなどだけでなく、その周辺の要素も含めて、完全なアプリケーションを構築することが必要である。本節では「住宅価格予測」装置の最も重要な要素をいくつか挙げて、何を使ったか明示した。

10.5.9 E-1: 運用時品質の維持性

定義

運用時品質の維持性とは、運転開始時に満足した内部品質が運転中も維持されることを

[10] 2019 年頃の状況に基づく。

意味する。言い換えると、システム外の運用環境の変化に内部品質が十分に対応でき、訓練した機械学習モデルが変化しても、不必要な品質劣化を引き起こさない。

運用開始時の品質が運用期間中も維持されているかを確認するためには、機械学習に基づくシステムや機械学習要素の挙動を継続的に監視する必要がある。

運用時品質の維持性のための一般的な分析プロセスまたは構造は以下の通り。

- 精度(KPI)モニタリング
- モデル出力モニタリング
- 入力データの監視

精度(KPI)モニタリング

精度モニタリングは、訓練した機械学習モデルの精度を直接測定することである。このモニタリングは、精度の算出に必要な、訓練済み機械学習モデルの推論結果に対する正解の収集方法に応じて、いくつかのパターンに分けられる。

モデル出力モニタリング

モデル出力モニタリングはさらに、医療診断のように出力推論ごとに専門家がチェックする場合と、一定期間後にすべての推論を一括してチェックする場合に分けられる。

入力データモニタリング

入力データモニタリングとは、訓練済み機械学習モデルによる推論結果のモニタリングと、その入力データのモニタリングを指す。このモニタリング方式は、住宅価格予測データの場合、人手で行うことができる。品質が低下した場合の対処方法を確認する必要がある。

おわりに

この内部品質については、特に結果を示していないが、機械学習技術の整備は、精度と頑健性の両面でモデルの改善に役立つ。

11 自動搬送車

本節では自動搬送車の業務要件記述だけを、他の事例よりも詳細な用途や環境条件の仕様を示した例として紹介する。

11.1 製品名

機能安全搬送車 DRC-X

11.2 ユースケース

この搬送車では、以下のユースケースを想定している。

1. サービス提供者はあらかじめ専用のソフトウェアにより、施設内の地図情報および目的地群を入力しておく。
2. 使用者は荷台に荷物を積載し、必要なら目的地を選択し、発進操作を行う。
3. 目的地の指定はスケジュールに従うか、制御装置の画面を用いた使用者の選択操作による。搬送を管理する外部システムからの指令を受領することも考えられる。
4. 周囲の人に発進を通知する警告音を発呼したのち発進した搬送車は自動的に目的地へ走行する。このときリアルタイムの監視を必要としない。
5. 搬送車の走行速度には上限を設定し、状況に応じて上限値は変化する。
6. 加減速は荷崩れなどの安定不安のない程度の加速度とする。
7. 搬送車は走行にあたり、障害物は迂回する。また、進行方向に障害物が接近し衝突が予見されると速度上限を制限することで減速する。ただし、人間の近くを通過する場合は上記にかかわらず減速する。なお、障害物が人と人以外のどちらかはっきりしない場合は人だと想定してふるまう。
8. 障害物と接触間際まで近接した場合は停車する。近接停車後、障害物が充分に離れた場合は一定時間後警告音を発呼し、速度上限を低速に限った状態から徐々に制限を解除しつつ発走する。
9. 障害物に近接停車した後一定時間経過してなお障害物が近接したままの場合であっても、障害物のない進行方向があれば、周囲の人に発進を通知する警告音を発呼し、速度を微速に制限し、障害物のない方向へ進行することで脱出を試みる。脱出後は一旦停車して、周囲の人に発進を通知する警告音発呼後通常の走行に移行する。
10. 目的地に到着すると、搬送車は停車して使用者に到着を通知する報告音を発呼する。使用者は荷物を車体からおろす。

11. 非常停止スイッチが車体外部にあり、押下によって確実に停車する。この際サービス提供者へ警告を通達する。また、一旦押下した非常停止スイッチを元に戻してもサービス提供者が解除操作を行うまで停車状態を維持する。
12. 使用者の指示やサービス提供者の遠隔指令またはスケジュールに従い、搬送車は待機位置へ自動的に移動する。待機位置では非接触の充電がおこなわれる。
13. サービス提供者は定期的な点検を実施する。
14. サービス開発者は一定期間にわたり、交換用消耗部品を供給し、また修理サービスをおこなう。
15. 搬送車が完全な位置喪失や故障などの異常を検知した場合は停車し、サービス提供者へ警告を通達する。
16. 長時間の停車など異常がある場合は、サービス提供者が現地へ赴き、あるいは遠隔で対処する。
17. サービス提供者は車両搭載のカメラ画像および位置情報を遠隔で得ることができ、走行を直接指示することができる。ただし、障害物の近接判定を無視した走行指示は遠隔ではなく臨場で行う必要がある。
18. 廃棄時はバッテリーを取り外し、産業廃棄物業者に処分を委託する。

11.3 ビジネス要件

11.3.1 背景

労働人口の減少にともない、商業施設や工場、物流現場での省人化のニーズに答える形での自動搬送車両の必要性が高まっている。より多様な現場での活用のためには、専用のレーンを必要とせず、人と活動空間を共有する方式が望まれる。この場合安全性は重大な課題であり、単に危害を伴う衝突をしないという程度にとどまらず、驚きや不安を与えるような挙動も抑制されるべきである。これは可能な限り高速に移動し輸送を達成すべきとする可用性の観点とは相反するものであるため、安全かつ人に不安を与えない範囲では速度を優先する方式が望まれる。

11.3.2 目的・目標

故障によるリスクも含め、安全を確保する

物体検出 AI によって周囲に人がいるかどうかを判定することで自動搬送車両の挙動を変え周囲の人の不安感を低減する

11.3.3 この製品のステークホルダー

このリファレンスガイドで考慮する、本製品のステークホルダーは以下の通りである。

- サービス開発者（搬送車を製造する）
- サービス提供者（現場にあわせシステムを設定し、メンテナンスと遠隔監視を行う）
- ユーザー（保有施設内で搬送車を利用する。発送側と受領側で同一である必要はない）
- 通行人
- 産業廃棄物業者

11.3.4 ステークホルダーの初期要求

- サービス開発者：搬送車の運用によって走行データを蓄積し今後の開発に活用したい。
- サービス提供者：少ない手間で役に立つ搬送システムを運用したい。
- ユーザー：安全な範囲で可能な限り高速に搬送してほしい。
- 通行人：安全かつ安心であってほしい。
- 産業廃棄物業者：危険なく廃棄物処分したい

11.3.5 ビジネス要件の詳細

開発する製品のビジネス要件は、以下の通りである。

前提条件

- 搬送車の走行環境は倉庫や工場など大規模な施設の屋内通路である。
- 車止めの障害物があるため階段やエスカレーター、段差への侵入の可能性はない。
- 同じ通路を台車や通行人、他の搬送車が通行する場合がある。
- 走行環境に動物や幼児など、悪意や無分別で搬送車に飛び込んでくるような生物はいない。
- 安全上の障害物はすべて搬送車の 2D LiDAR で検知可能な高さである。
- 照明などの環境条件はセンサーに適した状況である。
- 搭載する荷物は荷崩れや漏出の恐れがない。
- 物体検出 AI は NVIDIA GeForce RTX2060 Mobile 相当の計算リソースにおいて 30fps 以上の実時間処理能力が必要である。このため、DarkNet または類似のアーキテクチャに基づくネットワークモデルを用いる。
- 物体検出 AI モデルは汎用の学習済みモデルを使用する。

依存事項

- 異常時にサービサーを呼び出すための通信手段が必要である。
- 搬送先の指定を外部システムに依存する場合は、その API への対応が追加で必要である。

制約事項

- 走行速度上限の最大値は道交法の電動車いす等の規定にもとづき時速 6km とする。
- JIS D 6802 無人搬送車の規格に基づき、自律走行中は警告音を継続的に発呼し、旋回・後退時はウィンカーや後退音などで周囲に知らせる。

機能要件

- 通行人への危害を伴う衝突はもちろん、圧迫や擦過などのわずかな危害もないこと。
- 車体の押しやすい位置に非常停止ボタンを設け、押下時は完全に停車すること。
- 物体検出 AI の判定結果が誤っていたとしても、障害物および通行人への危害を伴う衝突のないこと。
- 車両の二次元 LiDAR および走行制御系は安全関連システムとして構築され、単一故障を検知すること。
- 車両は障害物検知のための 2 次元 LiDAR を備え、自己位置を推定して自律的に進行方向を決定して走行すること。
- 車両は進行方向と速度によって障害物への対処領域と領域ごとの速度上限を定め、より近い領域に障害物を検知するごとには段階的に減速することで衝突前に停止すること。
- 車両は物体検出 AI のための、カラー画像および深度情報を取得可能なカメラを備えること。
- 物体検出 AI は周囲のオブジェクトに対しインスタンスセグメンテーションを行う。ラベル出力は「人間」「非人間」「不明」の 3 カテゴリーに集約されること。
- 一般に学習データセットのラベル出力は細かく分類されるため、上記 3 カテゴリーに集約するためのフィルターをネットワークの出力段に追加で備えること。
- 物体検出 AI の出力と LiDAR の点群情報から、障害物の対処領域に侵入した物体のラベルを得ること。
- 障害物の近傍を通過する際、「非人間」と確信できなければ、衝突の恐れがなくとも減速して通過すること。
- 物体検出 AI の学習データセットは人物のほか、想定環境に予想される物品（机、椅子、箱、台車、プリンタまたはコピー機、棚など）を含むこと。
- 学習データセットはデータセット作成者以外の機関によって妥当性の評価を受けていること。
- 床、壁面、天井など、同一平面で広範囲に存在する物体については物体検出 AI ではなく、点群データからの平面抽出によって判断すること。

非機能要件

- サービス提供者は異常時に速やかに対処すること
- ユーザーは使用に当たり研修を受け、操作に習熟すること
- ユーザーは万一に備え傷害保険に加入すること
- センサーやカメラが収集した情報による合理的ではないプライバシー侵害のないこと。

考慮しない事項

- 搬送車への人の搭乗は想定しない。
- 搬送車が走行困難なほど混雑した状況は想定しない。

リスクと懸念事項

- 安全関連系の複数同時故障による異常動作のため危害を生じる恐れ。
- ２次元 LiDAR で検出困難な段差からの落下や、検出困難な人あるいは障害物への衝突により危害を生じる恐れ。
- 同種車両の交通集中による渋滞ないし通路封鎖の恐れ。
- 走行に伴い汚損、病原、火災などの危険源を拡散する恐れ。
- サイバー攻撃による、情報の漏洩や、偽装した走行指令や、異常動作が生じる恐れ。

11.3.6 外部品質に関する要求事項

開発する AI ソフトウェアに期待される品質要求レベルを、3 つの主要な外部品質の観点から以下に示す。

安全性

- 本製品の走行出力は衝突によって重症ないし死亡に至る身体的傷害の危険性がある。
- 環境への衝突により設備の破壊などの経済的損失を生じる恐れがある。
- 充電設備の不適切な取り扱いによって火災などの危険性がある。

パフォーマンス

- 搬送車システムは、利害関係者が合意した KPI 指標のしきい値を満たすべきである。
- システム全体として、処理速度と安全性のバランスがとれていることが求められる。
- 2 次元 LiDAR は SIL2 相当の機能安全に対応し、最大速度で 3 秒以上の走行距離に相当する安全防護領域を設定可能であること。またその 3 倍以上の距離までの点群情報を取得可能であること。
- 電源遮断時に制動状態となる電磁ブレーキを備えること
- 物体検出 AI は､1 秒間に 30 フレーム以上の物体検出が可能な処理能力を有すること。
- 物体検出 AI のラベル出力は、人間に対し「非人間」との回答が 10%未満、非人間に対し「非人間」との回答が 50%以上であること。
- 物体検出 AI において、車椅子などを使用中の人物も「人間」と判定しうる学習データが用いられること。

公平性

公平性に関する要求は現時点では明らかではない。

11.3.7 外部品質のレベルを定義する

本搬送車のうち、物体検出 AI を除く機能安全レベルは IEC61508 で定める SIL 2 を想定する。

MLQM ガイドラインに基づき、物体検出 AI に求められる外部品質のレベルを以下に示す。

外部品質	補足説明	想定される深刻度	実現すべきレベル
安全性	人的リスクに対する AI 安全レベル	物理的なダメージは想定していない。	AISL 0
	経済的リスクに対する AI 安全レベル	軽微な利益損失、人による監視で回避不可能	AISL 1
パフォーマンス	一般的 AI 性能レベル	KPI は事前に特定されるが、各 KPI の閾値は他の要因によって変動する可能性があり、ベストエフォートで提供される	AIPL 1
公平性	製品・サービスの公正さに関する明確な要件はない		AIFL 0

11.4 おわりに

本事例は、ユーザーのニーズと期待、このソリューションの背景にある目的、展開の成功に 影響を与える可能性のある高レベルの制約など、ビジネスの観点から説明されている。ここでは、自動搬送車両に関するビジネス要求を導き出した。このレポートは、PoC 終了の段階に関するスナップショットであり、開発者が次の段階で内部品質を評価するのに役立つ。

付録

A. ビジネス要件記述

各アプリケーション事例の冒頭には、架空の AI 製品に対する仮想的なビジネス要件を記述した。製品の開発者は通常、製品の開発を望む主体（サービス提供者／所有者）から、こういった形式化された要件を受け取る。本リファレンスガイドでは、事例の執筆者が自分で要件を例として冒頭の節で設定し、以降の製品開発プロセスではその設定に基づいて内部品質を評価できるようにした。

本文書に示したビジネス要件は必ずしも現実のシナリオを反映していない。以下では、本文書に示したビジネス要件について、目的、適用対象、および制約事項を説明する。

A.1　ステークホルダーの選択

ビジネス要件は、製品のステークホルダーのニーズから生まれるとされている。そのため、製品のステークホルダーをすべて特定すれば、それだけさまざまなビジネス要件の視点が得られる。

しかし、MLQM ガイドラインは、主に安全性、性能、公平性に関連する要件に関心を持つステークホルダー向けに書かれている。本リファレンスガイドも同じ意図の下で、製品の機能及び非機能要件に直接影響を与える要求事項を持つ、最も関連性の高いステークホルダーを挙げている。

A.2　繰返しによる調整

一般に、製品やサービスのビジネス要件は、通常、サービス提供者（製品等を開発してほしい主体）と開発者（製品等を開発する主体）の間で何度も議論・調整を繰り返して設定されることが多い。ビジネス要件記述を作る際もそういう手順を繰り返すが、本リファレンスガイドではその狙いに即して簡単に、最後に決まったビジネス要件のみを記述している。

A.3　視点の選択

通常のビジネス要件文書には、製品ライフサイクルの各段階に関連するさまざまな視点が含まれている。例えば、ビジネス要件記述書（BRD）の重要な要素である制約事項や依存事項には、財務やスケジュールに関する要件など、しばしば非技術的な視点が含まれている。これらは本リファレンスガイドで扱う事項の範囲外であるため、ここで示した BRD ではそれらに言及していない。実際の事例では、BRD にはこのような要件も含まれ、それに応じ

た管理を行う。

A.4　追加システムの要件

製品・サービス全体が AI モジュール以外の要素を持つことも多い。自動運転車の場合、ビデオストリームをキャプチャする単眼カメラや、ビデオから画像を抽出するシステムなどがその例である。通常、BRD には、それらの追加部分に関する要件も含まれている。しかし、このリファレンスガイドは、システムの AI 部分の品質のみを評価することを目的としているので、ここで扱う BRD では、AI システムの開発に密接に関係する要件のみを掲載している。

A.5　フォーマットの選択

実際の現場では、ビジネス要件書(BRD)、機能要求仕様書(FRS)、ソフトウェア要求仕様書(SRS)を個別に作成することが行われている。簡略化のため、ここに示す BRD は従来の FRS や SRS の要素を含むことがある。ここでの目的は、これらの文書の構造のお手本を示すことではなく、むしろ MLQM ガイドラインをこれらの文書にどう織り込むことができるか、また、サービス提供者が期待する品質目標をより明確に表現することにガイドラインがどう役立つかを例示することにある。

製品に期待される品質要求が定義できたら、開発時にどの点は完全達成必須で、どの点はできる限りの解決策を達成すれば十分なのかが明らかになっているはずである。この定義は、MLQM ガイドラインで言及されている外部品質特性レベルを用いて明確に表現することができる。このため、BRD の末尾にはガイドラインの外部品質軸で要件を表現し、開発者が明確かつ包括的に理解できるようにした。

ビジネス要件文書において、外部品質特性レベルを特定することは必須ではない。しかし、こうすれば、サービス提供者と開発者の双方に、達成すべき品質目標に対する期待を疑問の余地なく明らかにすることができる。

B. Surprise Adequacy

Surprise Adequacy [19]は、データセット内のデータの品質を調査する手法である。Surprise Adequacy は、訓練用データに対するバリデーション用データの振舞いの違いを定量的に測定する妥当性指標を定義する。測定には、ニューロンの活性化状況を把握するために**活性化トレース（AT）**を用いる。ニューラルネットワーク N を構成するニューロンの集合を $N=\{n_1, n_2, ...\}$とし、入力の集合を $X=\{x_1, x_2, ...\}$とする。入力 x に対するニューロン n の活性化値は、$a_n(x)$と定義される。ニューロンの順序付き集合 $N \subseteq N$ に対して、$a_N(x)$ は活性化値のベクトルを示し、各要素は N の個々のニューロンに対応する。$a_N(x)$ のカーディナリティは$|N|$に等しい。 $a_N(x)$ が N 中のニューロンに対する x の活性化トレースである。したがって、入力集合 X に対する活性化トレースは $A_N(X)=\{a_N(x)/x \in X\}$ と定義できる。

Surprise Adequacy は、まずすべての訓練用データの活性化トレース $A_N(T)$ を求め、その後、バリデーション用データから新たな入力 x の活性化トレース $A_N(x)$ を計算し、最後に、 $A_N(x)$ と $A_N(T)$ を比較することで得られる。比較する仕組みは色々あるが、本リファレンスガイドでは**距離に基づく** Surprise Adequacy（DSA）のみを使用する。今後、Surprise Adequacy の他の比較メカニズムを使うことも考えられる。なお、Surprise Adequacy, 活性化トレース、DSA について詳しく説明することは、このリファレンスガイドの目的ではない。

DSA は、新しい入力 x の AT と訓練中に観測された AT のユークリッド距離を使って定義されている。

$$\boldsymbol{dist_a = \|a_N(x) - a_N(X_a)\|} \tag{37}$$

$$\boldsymbol{dist_b = \|a_N(X_a) - a_N(X_b)\|} \tag{38}$$

$$\boldsymbol{DSA(x) = \frac{dist_a}{dist_b}} \tag{39}$$

さらに、DSA_1, DSA_2, DSA_3 の 3 つの DSA を定義する。また DSA_0 は先に説明した元の DSA とする。これらの新しい DSA は、データセットのコーナーケースを検出するために使用する。図 101 にこれらの DSA の違いを示す。DSA_1 は x の属するクラスでの新奇性と他のクラスでの新奇性を比較する。 $dist_a$ は DSA_0 と同じである。

$$\boldsymbol{x_b = \underset{c_{x_i} \in \{C - c_x\}}{\operatorname{argmin}} x e^{-x^2} \|a_N(x) - a_N(x_i)\|} \tag{40}$$

$$\boldsymbol{dist_b = \|a_N(x) - a_N(x_b)\|} \tag{41}$$

DSA_2 はテスト用データ $\boldsymbol{x}$ を全クラスのデータと比較する。

$$\boldsymbol{m} = \frac{\boldsymbol{1}}{\boldsymbol{k}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{k}} \boldsymbol{x_i}, \{\boldsymbol{x_i} | \boldsymbol{c_{x_i}} = \boldsymbol{c_s}\} \tag{42}$$

$$\boldsymbol{dist_a} = \|\boldsymbol{a_N}(\boldsymbol{x}) - \boldsymbol{a_N}(\boldsymbol{m_a})\| \tag{43}$$

$$\boldsymbol{dist_b} = \|\boldsymbol{a_N}(\boldsymbol{x}) - \boldsymbol{a_N}(\boldsymbol{m_b})\| \tag{44}$$

ここで、m_a はクラス c_a の中心点（$c_a = c_x$）、m_b はクラス c_b（$c_b \in \{C - c_x\}$）の最近接中心点を表す。

DSA_3 はデータ $\boldsymbol{x}$ の近傍領域の中心と k 近傍領域を比較する。$dist_a$ と $dist_b$ は DSA2 と同じである。

$$\boldsymbol{m} = \frac{\boldsymbol{1}}{\boldsymbol{k}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{k}} \boldsymbol{x_i}, \{\boldsymbol{x_i} | \boldsymbol{c_{x_i}} = \boldsymbol{c_s} \,\&\, \boldsymbol{x_i} \in \boldsymbol{N_k}(\boldsymbol{x})\} \tag{45}$$

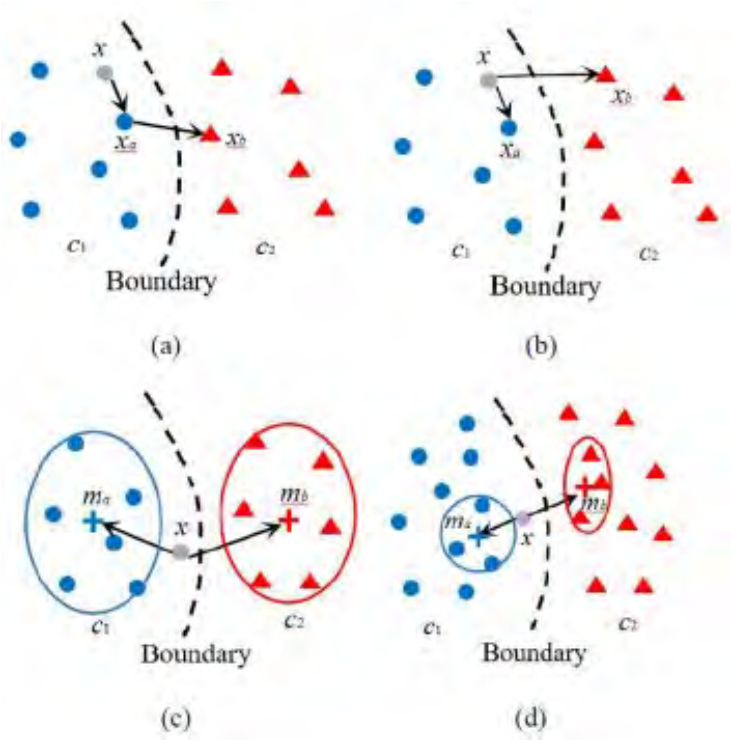


図 101 4 種の DSA の図:

元の *DSA*$_0$ (a), *DSA*$_1$ (b), *DSA*$_2$ (c), *DSA*$_3$ (d)

本文中の実験では、BDD100k データセットから、距離が 1 以上あるコーナーケースを削除した。その狙いは、訓練用データセットのコーナーケースを検出して取り除き、すべての検出モデル Yolov3 + ASFF, Yolov4, Yolov5, Fast R-CNN, MobileNetv2 を再訓練することにあった。

C. 1 ピクセル変更

文献には、敵対的データを生成するために広く使われている 3 つの距離メトリクスがあり、これらはすべて L_p ノルムである。

L_p 距離は $||x - x_0||_p$ と書かれ、p ノルム $||\cdot||_p$ は次のように定義される。

$$\|v\|_p = \left(\sum_{i=1}^{n} |v_i|^P\right)^{\frac{1}{p}} \tag{46}$$

1. L_0 距離は、$x_i \neq x'_i$ となるような座標 i の数を測定する。したがって、L_0 距離は、画像内で変更されたピクセルの数に相当する。

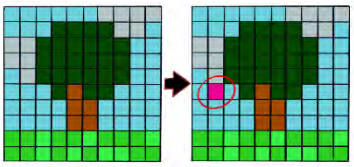


図 102 L_0 画像の例

2. L_2 距離は、x と x' の画像間の標準ユークリッド（二乗平均平方根）距離を測定する。L_2 距離は、多くのピクセルに小さな変更があっても小さいままであることがある。



図 103 L_2 画像の例

3. L_∞ 距離は、いずれかの座標に対する最大変化量を測定する。

$$\|x - x'\|_\infty = max(|x_1 - x'_1|, \dots, |x_n - x'_n|) \quad (47)$$

FGSM は敵対的データを生成するために L_∞を使用する。FGSM (Fast Gradient Sign Method) の定義はその名前も説明している。損失関数の勾配 (gradient) であり、摂動の大きさが L_∞で制限されているため、摂動の方向が勾配の符号 (sign) になる。




図 104 L_∞画像の例

頑健性の評価には、最大安全半径 (MSR) がよく使われる。画像 A と訓練済みモデル (分類器) f の MSR は、以下のような距離である。

$$MSR(A, f) = \max\{|A - A'|_p \, | f(A) = f(A')\} \quad (48)$$

言い換えれば、MSR は最も近い敵対的データまでの距離である。

本リファレンスガイドでは BDD100k について、ラベルごとに MSR を推定した。しかし、すでに L_∞を用いた既存研究があるため、L_0 や L_2 を用いて敵対的データを生成することを試みた。L_2 を用いた手法の開発は成功しなかったが、L_0 を用いて敵対的データを生成することができた。つまり、*p=0* での敵対的データの生成だけが成功した。L_0 は、画像にノイズを加えて画像上のピクセルを変化させるものである。つまり、1 ピクセル攻撃とは、L_0 に基づく攻撃である。

D. 評価の要約

本節では、本ガイドの最初の 4 つの例について、それぞれ実施したアセスメントの要約を表にしたものを示す。この表が列挙した事項には、言及されているが部分的にしか実施されていない、あるいは現時点では実施されていない評価も含まれている。完全には実施されなかった評価の重要性が低いわけではない。完全に実施されなかった理由は様々であり、多くの場合、適切なデータやその他のリソースが入手できなかったためである。

D.1 自動運転車

自動運転車のための物体検知と場面識別

機能要件

- 天気や道路状況その他の特徴に基づいて、現在の運転状況を識別する。
- 気候条件、交通条件、道路条件、時間帯、照明条件など、あらゆる条件下で、歩行者などの人、車、信号、標識、バス、バイクなど、運転中に日常的に遭遇する物体を検出する。

非機能要件

- AIモジュールは、FPS（毎秒処理フレーム数）で与えられる最低限の速度での処理において、必要な精度を維持する必要がある。
- AIモジュールは、自動運転車が走行する米国の都市部や郊外の交通ルールや状況に最も適合している必要がある。
- AIモジュールは、すくなくとも比較的低い走行速度では十分に動作すること。
- 物体検知と場面識別の機能は、稀な運転状況においても頑健である必要がある。

開発準備段階	内容	実施済	未実施	一部実施・実施中
予備的分析	技術仕様	学習の種類、機械学習アルゴリズム、両タスクに使うモデル候補		
	安全仕様	ASILとAISLやAIPLの対応付け		
	KPI仕様	物体検知にはmAPとFPS 場面識別には正解率	F1スコア	
PoCフェーズ	既存データセットの予備調査	データセット13件を調査		
	選択したデータセットの初期分析	属性とその値、正解ラベル、構造、分布に関する一般的情報		
	候補モデルの事前訓練	物体検知の訓練(データ7万件) 場面識別の訓練(データ2万件)	場面識別の訓練(データ7万件)	
	候補モデルのバリデーション	物体識別のバリデーション(データ1万件) 場面識別のバリデーション(データ1万件)	F1スコア、ロスカーブ(両タスク共)	
	PoCフェーズで得られた知見	PoCから内部品質評価を伴う開発への移行		
内部品質	**内容**	**実施済**	**未実施**	**一部実施・実施中**
A1 問題領域分析の十分性	新たな問題領域の提案	属性12個からなる新たな問題領域		
	データセットとの比較	既存の領域と提案した領域の比較		
	データセットを適合させるための修正	✓		
	最終的な問題領域	✓		
	….			
A2 データ設計の十分性	属性のあり得る組合せの個数の評価	属性7個のより小さいデータセットについて評価(注釈数約2千)	全属性＋全データ	
	ペアワイズ分析	21通りの組合せのうち、明度＋信号についての評価と分布		
	多属性の組合せの分析		(3個以上の属性の組合せ)	
	不健全なケース	(ペアワイズの)不健全なケースの設定例	(3個以上の属性の組合せ)	
	安全上重大な／高リスクケース	(ペアワイズ+3属性の組合せの)高リスクケースの設定例		
	….			

B1 データセットの被覆性	大局的被覆性	✓		
	局所的被覆性	定義した不健全なケース 定義した安全上重大なケース		
	被覆性を改善する技術(開発ステージ用)		ニューロン・カバレッジ、Surprise Coverage	フレーム抽出、データ拡張
B2 データセットの均一性	大局的均一性	一般的分布の評価例 似ているがより小さな別のデータセットとの比較例		
	局所的均一性	組合せケースの分布の評価例 似ているがより小さな別のデータセットとの比較例		
	…			
B3 データの妥当性				
	….			
C1 機械学習モデルの正確性	全般的な正確性	場面識別では正解率、物体検知ではmAP	F1スコア	
	特定のケースでの正確性			組合せて使うためのmAP
	正確性向上技術(開発段階向け)	小さなバウンディングボックスの削除 属性別の訓練		コーナーケースの削除
	….			
C2 機械学習モデルの安定性	汎化性能の評価	物体検知タスクに関する、別のデータセット(nuImage)での確認	場面識別タスクに関する別のデータセットでの確認、 両タスクに関するロスカーブ	
	ノイズや敵対的データに対する頑健性の評価	物体検知モデルに関する敵対的攻撃についての確認(Surprise Adequacy, FGSM, 1ピクセル)	場面識別モデルに関する敵対的攻撃についての確認 (Surprise Adequacy)	場面識別モデルに関する敵対的攻撃についての確認(ニューロン・カバレッジ)
	頑健性向上技術(開発段階向け)		生成したデータによる再訓練	
D1 プログラムの信頼性	アルゴリズムの正しさ	PoCで重みの出典を明示		
	オープンソース要素の健全性	プログラミング言語 (Python) 機械学習フレームワーク (TensorFlow) その他ライブラリ (Numpy,SciPy, Pandas,Matplotlib)	コンテナ環境 (Docker等)への展開可能性	
	訓練および運用環境におけるハードウェアの信頼性			

	メモリ使用量の健全性	モデルのアーキテクチャと重み ソースコード 入力データ 演算ユニットの仕様		
	訓練時間や推論時間の効率		物体検知と場面識別タスクでの確認	
	…			
E1 運用時品質の維持性	精度のモニタリング		✓	
	モデル出力と入力データのモニタリング		✓	
	KPIのモニタリング		✓	
	…			

D.2　金属鋳物の外観検査

金属鋳物の外観検査

機能要件

- 鋳造品の不良を認識できること

非機能要件

- 精度が目標とする品質標準(業界標準等）を満足する(精度がX%を超える等)ものであること。
- 認識プロセスは、誤認識率やヒット率など、様々な誤差基準を満たし、また、目標とする品質標準が定める閾値に達すること。
- 所定の範囲内の様々な解像度の画像に対して頑健なシステムであること。
- 様々なメーカーのカメラで撮影された画像に対し頑健なシステムであること。

内部品質	内容	実施済	未実施	一部実施・実施中
A1 問題領域分析の十分性	AIモデルの要件	欠陥検知のための2値分類問題。 欠陥種別判定のための多値分類問題。	欠陥位置の特定	
	データセットの選定	金属鋳造データセット		
	データ型の要件	グレー画像	2値正解情報，マルチチャネル画像	
	属性の選択	(輝度、コントラスト、露光)		✓
	属性の領域	✓		✓
	KPIの定義と選択	(正解率、適合率、再現率)	Fスコア等	
	KPI要件		✓	
	…			
A2 データ設計の十分性	個別の属性の分析	✓		
	ペアワイズ分析	✓		
	多属性の組合せの分析			(3属性の組合せ)
	コーナーケース			(定義と検出)
	高リスクケース		✓	
	…			
B1 データセットの被覆性	大局的被覆性	✓		
	局所的被覆性			(いくつかの被覆性を定義して検討した)
	ケースの被覆性	✓		
	属性に基づく被覆性	✓		
	ニューロンベースの被覆性		✓	
	Surprise Coverage			✓
	…			
B2 データセットの均一性	大域的均一性			✓
	局所的均一性			✓
	ケース間にわたる均一性	✓		
	…			
B3 データの妥当性	データ拡張			✓
	敵対的データ生成		✓	
	メタモルフィックデータ生成		✓	
	コーナーケース生成		✓	
	…			

C1 機械学習モデルの正確性		全般的な正確性	(正解率)		
		観測された全正例に対する正例予測の正確性	(再現率)		
		予測された全正例に対する正例予測の正確性	(適合率)		
		ケースごとの正確性			✓
		コーナーケースデータ検出における正確性	✓		
		…			
C2 機械学習モデルの安定性		頑健性測定	(局所的頑健性の測定)		✓
		頑健性改善			(コーナーケースデータを考慮して)
		敵対的テスト		✓	
		データ変更テスト		✓	
		…			
D1 プログラムの信頼性		データの信頼性評価		(時追欠陥のデータ)	
		プラットフォームの信頼性評価		(Python, TensorFlow, PyTorch)	
		実行環境の信頼性評価		(Windows, MacOs)	
		物理的運用環境の信頼性評価		✓	
		…			
E1 運用時品質の維持性		運用性評価		✓	
		オンライン学習		✓	
		リスク管理		✓	
		…			

D.3　郵便番号の分析

郵便番号の分析のための手書き文字認識						
機能要件						
・　手書き文字を認識する						
非機能要件						
・　精度は満足な正解率レベルに達すること						
・　高速に推論できること						
・　一定の閾値を超えない範囲のノイズに対して頑健であること						
・　様々なスタイルの手書き文字を受け付けること						
内部品質			内容	実施済	未実施	一部実施・実施中
A1 問題領域分析の十分性			あり得る全てのクラスのデータ	MNISTデータセットを用いた。0から9までの数字について良い分布を備えている。	RGB画像。	
			特徴次元の定義	(位置、面積、長さ、明度、傾き、太さ、手書きのスタイル)		
			範囲内・範囲外の選択	(領域、大きさ、明度、傾き、太さ、手書きのスタイル)		
			…			
A2 データ設計の十分性			不健全な（起きるはずのない）ケースの特定	(該当なし)		
			…			
B1 データセットの被覆性			従来のデータカバレッジの特定	(特徴量面積の分布と被覆性、値レベルの被覆性、パターンレベルの被覆性、その他の変種)		
			Surprise adequacyに基づくカバレッジ	面積と長さの様々なクラスに関するsurprise coverageと精度		
			コーナーケースの特定	DSAに基づくコーナーケースの検出		
			特徴削減		特徴量に基づく変異検査	
			…			
B2 データセットの均一性			視覚的表現からデータ分布を把握	(グラフ表現)		
			データ均等性を計算	TPCov		
			データ収集の偏りを減らす		✓	
			データ拡張	(画像のコントラストの変更)		
			外部処理追加による特徴削減	(文字の位置)		
			…			
B3 データの妥当性						
			…			

C1 機械学習モデルの正確性	様々な精度指標とKPI	(混同行列、正解率、適合率、再現率、Fスコア)		
	モデルの動作定義とコーナーケースの検出		(入力データの優先度付け)	
	…			
C2 機械学習モデルの安定性	データの摂動に対する頑健性	(CNN-Cert, Fast-Lin)		
	モデルの摂動に対する頑健性	(変異頑健性)		
	…			
D1 プログラムの信頼性	プログラム＆オープンソースライブラリ	(Python, Tensorflow)		
	画像処理ユニット	✓		
	外部処理用ユニット	(数字の位置の修正)		
	メモリの使用状況		✓	
	時間コスト		✓	
	ソフトウェアのセキュリティ		✓	
	訓練環境と運用環境の違い		✓	
	…			
E1 運用時品質の維持性	精度(KPI)モニタリング		✓	
	継続的なデータ収集とラベリング		✓	
	モデルへの入力の新規性分析		✓	
	再訓練の必要性の分析		✓	
	モデル出力のモニタリング		✓	
	追加データセットの作成		✓	
	…			

D.4　住宅価格分析

住宅価格分析					
機能要件					
• AIモデルが、家の特徴に関する提供された情報をもとに、家の価格を推定する。					
非機能要件					
• AIモデルは、アイオワ州の住宅について良好な性能を発揮する。この地域の外の住宅は対象外である。					
• 安価なものから高価なものまで、幅広い価格帯の住宅を考慮する。					
• なんらかの特徴量の値が欠けていても、AIモデルは予測を継続する。					
• AIモデルは、非常に稀な特徴量を持つ住宅の価格推定に際しても頑健である。					
内部品質		内容	実施済	未実施	一部実施・実施中
A1 問題領域分析の十分性		問題領域の定義	(特徴量、データポイント、次元);		
		ありうる全ての価格帯のデータ	Kaggle house priceのデータセットを用いた。1460件のデータポイントがある。		
		適切な特徴量次元の選択	相関行列	(編集削減法、PCA)	
		範囲内・範囲外の選択	(GrLivArea , ExterQual)	他77件の特徴	
		不健全事例の特定			
		….			
A2 データ設計の十分性		各特徴量次元についてのデータ管理	カバレッジの確認		
		….			
B1 データセットの被覆性		各組合せのカバレッジ	特徴空間における分布		
		レアケース・コーナーケースの見極め	各領域が何らかのデータでカバーされているかどうかを確認する		
		特徴削減			
		….			
B2 データセットの均一性		各ケースに十分なデータがあるか	期待する分布と実際の分布の比較		
		…			
B3 データの妥当性					
		….			
C1 機械学習モデルの正確性		具体的なKPIの選択	(平均絶対対数誤差 MALE)	MSE, RMSE	
		データ分布の違いに関する性能比較	属性の分布の違いがモデル性能にどう影響するか		
		….			
C2 機械学習モデルの安定性		訓練とテストのロスカーブ	(ハイパーパラメータの調整、異なるアーキテクチャのモデル)		
		….			
D1 プログラムの信頼性		言語	プログラミング言語(Python)		
		フレームワーク	(Tensorflow, keras)	(Pytorch,Theano)	
		メモリ使用量		✓	
		メタパラメータ		✓	
		ハードウェア	CPU	TPU	GPU
		ソフトウェアのセキュリティ		✓	

			…			
E1 運用時品質の維持性			精度 (KPI) モニタリン		✓	
			モデル出力モニタリング		✓	
			入力データモニタリング		✓	

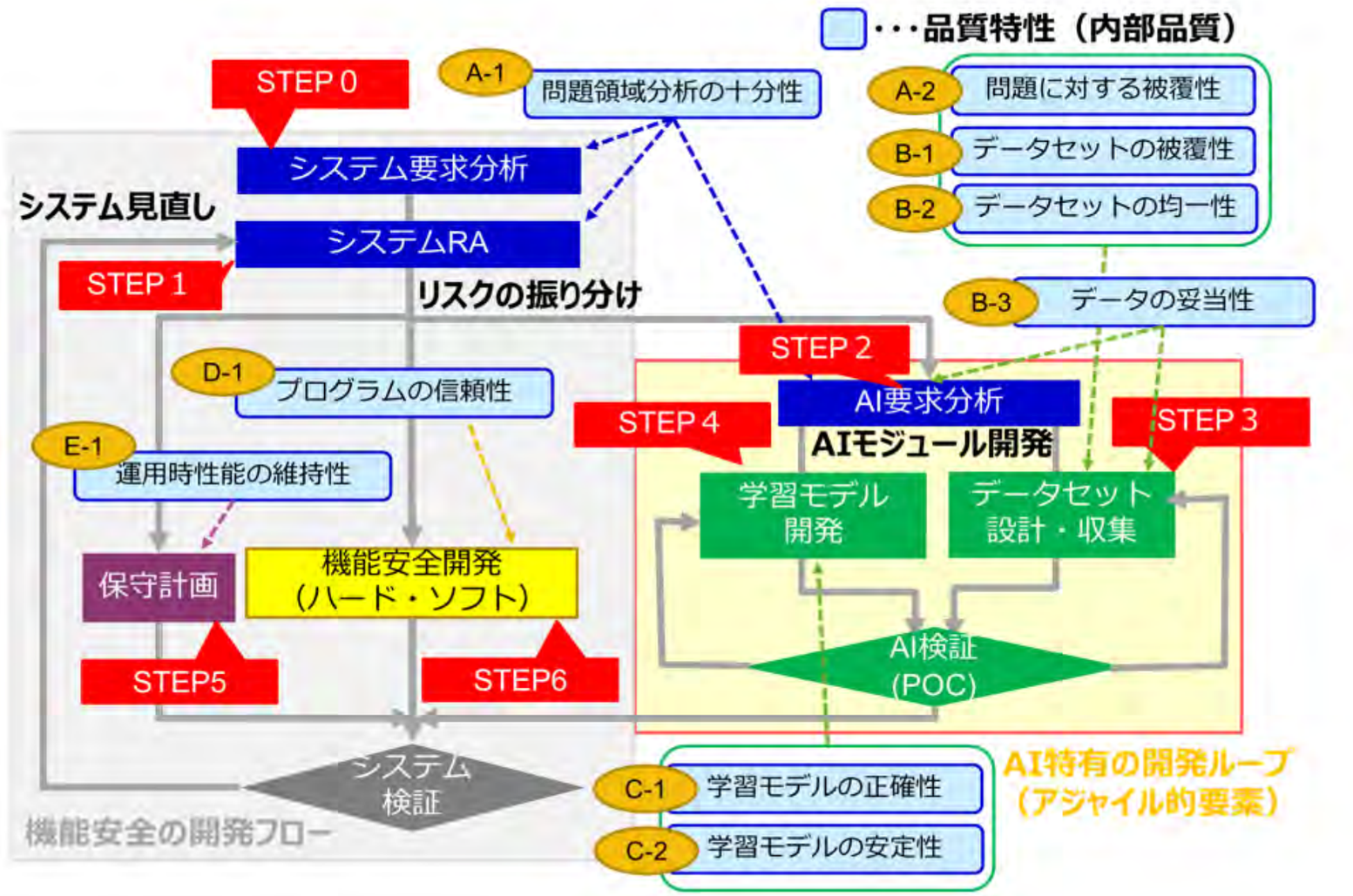
E. 品質アセスメントシート

6 章で紹介した品質アセスメントシートの未記入完全版を示す。また、一部のシートの使用例も掲載している。シートの使用方法については、6 章を参照されたい。

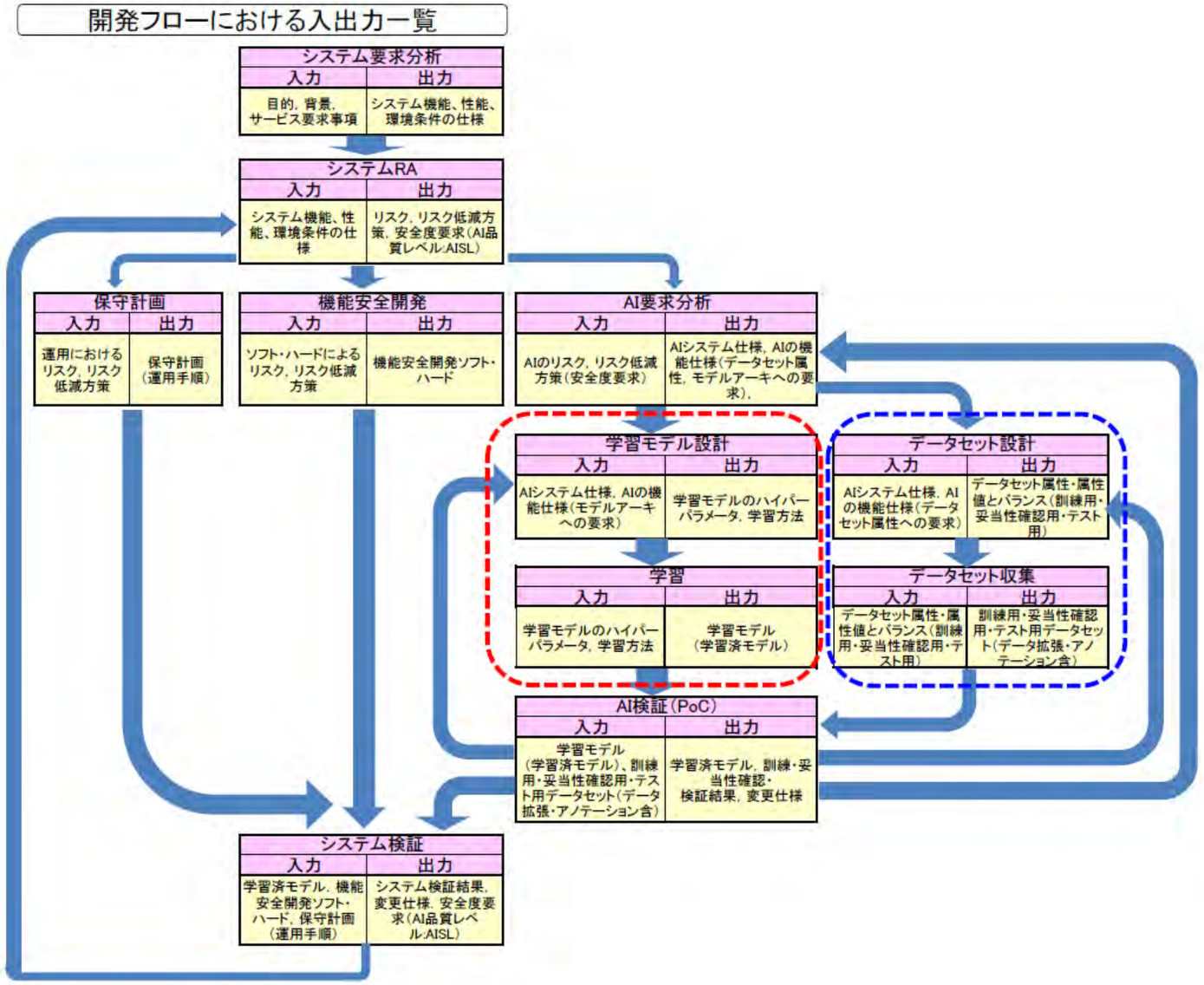
AI利用システム・品質アセスメントシート

STEP0	システム要求分析
STEP1	システム・リスクアセスメント
STEP2	AI要求分析
STEP3	データセット・アセスメント （データの妥当性アセスメント含む）
STEP4	機械学習モデル・アセスメント
STEP5	保守計画アセスメント
STEP6	機能安全プロセス管理アセスメント

Ver.2.9


・・・品質特性（内部品質）
STEP 0
A-1
問題領域分析の十分性
A-2
問題に対する被覆性
B-1
データセットの被覆性
B-2
データセットの均一性
システム要求分析
システム見直し
システムRA
STEP 1
リスクの振り分け
B-3
データの妥当性
STEP 2
D-1
プログラムの信頼性
STEP 4
AI要求分析
AIモジュール開発
STEP 3
E-1
運用時性能の維持性
学習モデル
開発
データセット
設計・収集
保守計画
機能安全開発
(ハード・ソフト)
AI検証
(POC)
STEP5
STEP6
システム
検証
C-1
学習モデルの正確性
C-2
学習モデルの安定性
AI特有の開発ループ
（アジャイル的要素）
機能安全の開発フロー

開発フローにおける入出力一覧



システム要求分析票

・・・PoC初期段階　・・・＋PoC最終段階/商品開発開始時　・・・＋商品開発完了時　・・・＋運用時

システム概要			
製品名		システム想定構成	
品　番			
目　的			

No.	ユースケース	内容	入力	出力	条件

1ページ

No	仕様分類	システム要求分析
1	機能	
2		
3		
4	性能	
5		
6		
7		
8	環境条件	
9		
10		

No	構成	
	分類	要素
1	ハードウェア	
2		
3		
4	ソフトウェア	
5		
6		

システム要求分析票の活用例

システム要求分析票

・・・PoC初期段階　・・・＋PoC最終段階/商品開発開始時　・・・＋商品開発完了時　・・・＋運用時

システム概要			
製品名	機能安全 搬送車 DRC-X （インテリジェント車いす）	システム想定構成	・使用方法：屋内搬送（手動／自動・切り替え型） ・本体(主部品)：電動車椅子×1台（無人制御可能） ・センサ：カメラ（画像に基づく人検出、および測域）×1台 レーザ・スキャナ（2D-LRF, Area OSSD）×3台 （左側/右側/背面） IMU×1台 ・通信：RS-232C（速度/加速度/エンコーダ値/バッテリ情報/制御入力情報） ・ユーザI/F：ジョイスティック（電動車いす標準）×1台, コントローラ（移動/停止/緊急停止ボタン）
品　番	－		
目　的	電動車いすに人が乗っていない時に、商品を屋内搬送するために安全自動搬送車として電動車いすを使う		

No.	ユースケース	内容	入力	出力	条件
1	初期設定	サービサーはあらかじめ専用のソフトウェアにより、施設内の地図情報および目的地群を入力しておく。	地図情報と目的地群	－	－
2	目的地の特定と発進	使用者は荷台に荷物を積載し、必要なら目的地を選択し、発進操作を行う。 目的地の指定はスケジュールに従うか、制御装置の画面を用いた使用者の選択操作による。	目的地と発進のトリガ	搬送車の発進	もし、システムが目的地を特定するのを外部システムに依存する場合は、APIの追加サポートが必要。
3	搬送の遠隔制御	搬送を管理する外部システムからの指令を受領することも考えられる。	遠隔操作コマンド	搬送車のアクション	－
4	走行	搬送車の走行環境は、倉庫や工場などの大規模施設における屋内通路とする。車両障壁などの障害物があるため、階段、エスカレーター、段差に入る可能性がない。同じ通路をカート、歩行者、他の輸送車両が使用することがある。	階段、エスカレータ、段差、を除く屋内環境。カート、歩行者、他の搬送車 。	搬送車走行	・走行環境には、悪意や無意味に輸送車両に飛び込むような動物や幼児などの生物はいない。 ・すべての安全障害物は、車両の2Dライダーで検知できる高さにある。 ・照明などの環境条件がセンサーに適していること。 ・積載貨物が倒れたり、漏れたりするおそれがないこと。 ・自律走行中は警告音を常時発し、旋回時や後退時にはウィンカーやバック音で周囲に知らせること。
5	障害物回避	搬送車は走行にあたり、障害物は迂回する。また、進行方向に障害物が接近し衝突が予見されると速度上限を制限することで減速する。ただし、人間の近くを通過する場合は上記にかかわらず減速する。なお、障害物が人と人以外のどちらかはっきりしない場合は人だと想定してふるまう.	障害物	人か人でないか	－
6	接近停止と解除制約	障害物と接触間際まで近接した場合は、停車する。近接停車後、障害物が充分に離れた場合は一定時間後警告音を発呼し、速度上限を低速に限った状態から徐々に制限を解除しつつ発走する。	障害物接近情報	速度制限命令	－
7	タイムアウトと通常走行への復帰	障害物に近接停車した後一定時間経過してなお障害物が近接したままの場合であっても、障害物のない進行方向があれば、周囲の人に発進を通知する警告音を発呼し、速度を徴速に制限し、障害物のない方向へ進行することで脱出を試みる。脱出後は一旦停車して、周囲の人に発進を通知する警告音発呼後通常の走行に移行する。 脱出後は一旦停車して、周囲の人に発進を通知する警告音発呼後通常の走行に移行する。	障害物の近くで停車した後、一定時間経過したこと。 障害物なしで進行可能な方向。	警告音, 走行制御 (停止および 通常走行の再開)	－

8	加減速	加減速は荷崩れなどの安定不安のない程度の加速度とする。	加速度	安定走行	･上限速度は 6 km/h（道路交通法における電動車いすのための上限速度に基づく）
9	目的地への到着	目的地に到着すると、搬送車は停車して使用者に到着を通知する報告音を発呼する。使用者は荷物を車体からおろす。	―	停止および警告音	―
10	緊急停止	非常停止スイッチが車体外部にあり、押下によって確実に停車する。この際サービサーへ警告を通達する。また、一旦押下した非常停止スイッチを元に戻してもサービサーが解除操作を行うまで停車状態を維持する。	非常停止スイッチを押す	停止	―
11	待機位置への移動	使用者の指示やサービサーの遠隔指令またはスケジュールに従い、搬送車は待機位置へ自動的に移動する。	使用者の指示, サービサーの遠隔指令, またはスケジュール	待機位置への移動	―
12	定期検査	サービサーは定期的な点検を実施する。	―	―	―
13	部品交換および修理	デベロッパは一定期間にわたり、交換用消耗部品を供給し、また修理サービスをおこなう。	―	―	―
14	異常検知と警告	搬送車が完全な位置喪失や故障などの異常を検知した場合は停車し、サービサーへ警告を通達する。	異常検知（例. 完全な位置喪失や故障）	停止、およびサービサーへの警告	･システムには、異常時にサービサーを呼び出す通信手段を備える必要がある。
15	サービサーの訪問、または遠隔保守	長時間の停車など異常がある場合は、サービサーが現地へ赴き、あるいは遠隔で対処する。	長時間の停車など異常	―	―
16	搬送車の遠隔走行	サービサーは車両搭載のカメラ画像および位置情報を遠隔で得ることができ、走行を直接指示することができる。	搬送車の走行指示	カメラ画像と位置情報	―
17	臨場走行	障害物の近接判定を無視した走行指示は遠隔ではなく臨場で行う必要がある。	障害物の近接判定を無視した走行指示	障害物の近接判定を無視した走行	―
18	ライフサイクルの終了	（18）廃棄時はバッテリーを取り外し、産業廃棄物業者に処分を委託する。	―	―	―

No	仕様分類	システム要求分析
1	機能	· The vehicle's 2D Lidar and driving control
2		· The vehicle shall be equipped with a two-
3		· The vehicle shall set an upper speed limit
4		· The vehicle is equipped with a camera
5		· When passing near an obstacle, if the vehicle is not convinced that it is "non-human," it will slow down and pass even if there is no threat of collision.
6	性能	—
7	環境条件	· Servicer shall take prompt action in case of
8		· Users should receive training and be familiar
9		· The user should have accident insurance in
10		· There shall be no unreasonable invasion of
11		· It is not assumed that people will be riding in
12		· It is not assumed that the transport vehicle

No	構成	
	分類	要素
1	ハードウェア	電動車いす（DRC-X）
2		LRF（レーザ・レンジ・ファインダ）
3		カメラ
4	ソフトウェア	YOLOv3
5		ROS
6		ubuntu 18.04 LTS

システム・リスクアセスメント票

・・・PoC初期段階　・・・＋PoC最終段階/商品開発開始時　・・・＋商品開発完了時　・・・＋運用時

システム概要			
製品名	0	システム構成想定	0
品　番	0		
目　的	0		

No	第1段階　リスクの抽出・見積り										第2段階　リスク低減策検討							
	使用ステップ	危険源(システムの部位)	危害モード	危害を受ける箇所	危害の発生状況(どのように危害が発生するか)	リスク推定				許容可否(○/×)	リスク低減方策(本質安全設計／防護安全設計／使用上の注意)	具体的な方策例	方策対象(AI／機能安全開発／保全)／(関連資料へのリンク)		リスク低減結果			許容可否(○/×)
						危害の重大性	危害の発生頻度	評価値	危害の算出根拠(関連資料へのリンク等)						危害の重大性	危害の発生頻度	評価値	

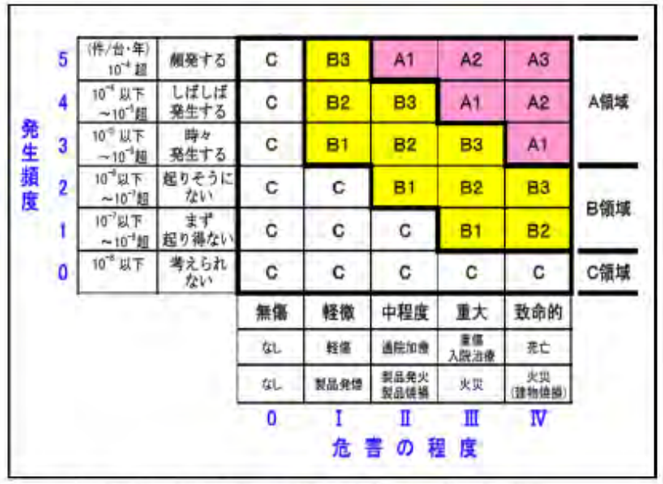
システム・リスクアセスメント票のリスクマップの一例

システム・リスクアセスメント票

・・・PoC初期段階　・・・＋PoC最終段階/商品開発開始時　・・・＋商品開発完了時　・・・＋運用時

システム概要			
製品名	0	システム構成想定	0
品　番	0		
目　的	0		

No	第1段階 リスクの抽出・見積り										第2段階 リスク低減策検討							
	使用ステップ	危険源（システムの部位）	危害モード	危害を受ける箇所	危害の発生状況（どのように危害が発生するか）	リスク推定				許容可否（O/×）	リスク低減方策（本質安全設計／防護安全設計／使用上の注意）	具体的な方策例	方策対象（AI／機能安全開発／保全）／（関連資料へのリンク）		リスク低減結果			許容可否（O/×）
						危害の重大性	危害の発生頻度	評価値	危害の算出根拠（関連資料へのリンク等）						危害の重大性	危害の発生頻度	評価値	



*本リスクマップは、リスク定義の一例であり，
この形式に限定されるものではない。
* "Applying the R-Map Method to Product Safety and Risk Management, Japan", は、
リスク管理方法の一つとして、ISO13077:2013(en)から参照されている。

AI要求分析票

・・・PoC初期段階　・・・＋PoC最終段階/商品開発開始時　・・・＋商品開発完了時　・・・＋運用時

システム概要				AI品質の要求レベル		
製品名	0	システム構成想定	0	外部品質	リスク回避性	AISL**/Lv*
品　番	0				AIパフォーマンス	AIPL**/Lv*
目 的	0				公平性	AIFL**/Lv*

No	要求内容（外部仕様）					前提条件（前処理・後処理）	教師あり／教師なし／強化学習	検討が必要な項目	属性（主要属性一覧）	根拠・理由	データセットへの要求（Policy）					機械学習モデルへの要求								構成ハード・ソフトへの要求（データセットと機械学習モデル以外）
	ユースケース（対象）	処理内容	入力	出力	リスクアセスメントNo.						過去の実績、POCでの知見	データセット属性（データの量，分布）	データ条件（データの質，数，サイズ・時空間制約，種別，汚染対策、メタデータの精度・ルール）	データ妥当性のPolicy	制約（データクレンジングなど）	過去の実績、POCでの知見	モデル精度（正解率，適合率，再現性，F値，など）				入力特性（空間的・時系列的）	出力特性（多クラス分類，信頼度情報有無，閾値，など）	制約（学習時間，ハイパーパラメータ対象，必要なリソースなど）	

データセット・アセスメント票

・・・PoC初期段階　・・・＋PoC最終段階/商品開発開始時　・・・＋商品開発完了時　・・・＋運用時

システム概要				AI品質の要求レベル		
製品名	0	システム構成想定	0	外部品質	リスク回避性	AISL**/Lv*
品　番	0				AIパフォーマンス	AIPL**/Lv*
目　的	0				公平性	AIFL**/Lv*

データ追加・拡張方法		アノテーション方法		公平性への対応方法	
内容	ツール名	内容	ツール名	内容	ツール名

＊「データ妥当性確認」シート参照

No	属性の抽出				オリジナル・データセットの構成											1回目データセット収集									
	データセット属性				訓練用データセットの構成			妥当性確認用データセットの構成			検証用データセットの構成			データの妥当性確認		データ追加・拡張の有無	拡張後の訓練用データセットの構成			拡張後の妥当性確認用データセットの構成			拡張後の検証用データセットの構成		
	中属性	小属性	属性値	対象	被覆性	分布[%]	データ数または量[件 or sec]	被覆性	分布[%]	データ数または量[件 or sec]	被覆性	分布[%]	データ数量[件 or sec]	データ	ラベリング(メタデータ)	データ数量[件 or sec](追加・削除・拡張)	被覆性	分布[%]	データ数または量[件 or sec]	被覆性	分布[%]	データ数または量[件 or sec]	被覆性	分布[%]	データ数量[件 or sec]
														妥当性確認No,＊	妥当性確認No,＊									

データの妥当性確認		1回目　確認結果														N回目　データセットの構成									
		検証条件					検証結果									訓練用データセットの構成			妥当性確認用データセットの構成			検証用データセットの構成			データの妥当性確認	
データ	ラベリング(メタデータ)	利用するデータセット Ver.	利用するMLモデル Ver.	訓練用プログラム Ver.	妥当性確認用プログラム Ver.	検証用(テスト)プログラム Ver.	属性値ごとの結果					属性ごとの結果(精度)[%]	全体の結果(精度)[%]	ツールによる分析結果(Pair-wise Analysisなど)	評価	被覆性	分布[%]	データ数量[件 or sec]	被覆性	分布[%]	データ数量[件 or sec]	被覆性	分布[%]	データ数量[件 or sec]	データ	ラベリング(メタデータ)
妥当性確認No,＊	妥当性確認No,＊						正解率[%]	適合率[%]	再現性[%]	F値	その他														妥当性確認No,＊	妥当性確認No,＊

N回目　データセット収集													N回目　確認結果																
データ追加・拡張の有無	拡張後の訓練用データセットの構成			拡張後の妥当性確認用データセットの構成			拡張後の検証用データセットの構成			データの妥当性確認		その他の改善ポイント（アノテーションの精度，敵対的データ，など）	検証条件					検証結果										
データ数量［件 or sec］（追加・削除・拡張）	被覆性	分布［%］	データ数または量［件 or sec］	被覆性	分布［%］	データ数または量［件 or sec］	被覆性	分布［%］	データ数量［件 or sec］	データ	ラベリング（メタデータ）		利用するデータセット Ver.	利用するMLモデル Ver.	訓練用プログラム Ver.	妥当性確認用プログラム Ver.	訓検証用（テスト）プログラム Ver.	属性値ごとの結果					属性ごとの結果（精度）［%］	全体の結果（精度）［%］	ツールによる分析結果（Pair-wise Analysis など）	評価			
										妥当性確認No *,	妥当性確認No, *							正解率［%］	適合率［%］	再現性［%］	F値	その他						

データセット・アセスメント票の使用例

データセット・アセスメント票

・・・PoC初期段階　・・・＋PoC最終段階/商品開発開始時　・・・＋商品開発完了時　・・・＋運用時

システム概要				AI品質の要求レベル		
製品名	0	システム構成想定	0	外部品質	リスク回避性	AISL**/Lv*
品　番	0				AIパフォーマンス	AIPL**/Lv*
目　的	0				公平性	AIFL**/Lv*

データ追加・拡張方法		アノテーション方法		公平性への対応方法	
内容	ツール名	内容	ツール名	内容	ツール名

＊「データ妥当性確認」シート参照

No	属性の抽出				オリジナル・データセットの構成											1回目データセット収集									
	データセット属性				訓練用データセットの構成			妥当性確認用データセットの構成			検証用データセットの構成			データの妥当性確認		データ追加・拡張の有無	拡張後の訓練用データセットの構成			拡張後の妥当性確認用データセットの構成			拡張後の検証用データセットの構成		
	中属性	小属性	属性値	対象	被覆性	分布[%]	データ数または量[件 or sec]	被覆性	分布[%]	データ数または量[件 or sec]	被覆性	分布[%]	データ数量[件 or sec]	データ	ラベリング(メタデータ)		データ数量[件 or sec](追加・削除・拡張)	被覆性	分布[%]	データ数または量[件 or sec]	被覆性	分布[%]	データ数または量[件 or sec]	被覆性	分布[%]	データ数量[件 or sec]
														妥当性確認No,＊	妥当性確認No,＊											
	人	明るさ		✔	6795	15	45174							1	1	-									
				✔	36445	81								1	1	-									
				✔	1934	4								1	1	-									
	自転車	明るさ	明るい		36	2	2287							1	1	-									
			普通		1939	85								1	1	-									
			暗い		312	14								1	1	-									
	車	明るさ	明るい		144	2	8606							1	1	-									
			普通		7403	86								1	1	-									
			暗い		1059	12								1	1	-									
	バイク	明るさ	明るい		26	1	2442							1	1	-									
			普通		2039	83								1	1	-									
			暗い		377	15								1	1	-									

データの妥当性確認		1回目　確認結果													N回目　データセットの構成									
		検証条件					検証結果								訓練用データセットの構成			妥当性確認用データセットの構成			検証用データセットの構成			データの妥当性確認	
データ	ラベリング（メタデータ）	利用するデータセット Ver.	利用するMLモデル Ver.	訓練用プログラム Ver.	妥当性確認用プログラム Ver.	検証用（テスト）プログラム Ver.	属性値ごとの結果					属性ごとの結果（精度）[%]	全体の結果（精度）[%]	ツールによる分析結果（Pair-wise Analysisなど）	評価	被覆性	分布[%]	データ数量[件 or sec]	被覆性	分布[%]	データ数量[件 or sec]	被覆性	分布[%]	データ数量[件 or sec]	データ	ラベリング（メタデータ）
妥当性確認No, *	妥当性確認No, *						正解率[%]	適合率[%]	再現性[%]	F値	その他													妥当性確認No, *	妥当性確認No, *

N回目　データセット収集													N回目　確認結果										
データ追加・拡張の有無	拡張後の訓練用データセットの構成			拡張後の妥当性確認用データセットの構成			拡張後の検証用データセットの構成			データの妥当性確認		その他の改善ポイント（アノテーションの精度，敵対的データ，など）	検証条件					検証結果							
										データ	ラベリング（メタデータ）		利用するデータセット Ver.	利用するMLモデル Ver.	訓練用プログラム Ver.	妥当性確認用プログラム Ver.	訓検証用（テスト）プログラム Ver.	属性値ごとの結果					属性ごとの結果（精度）[%]	全体の結果（精度）[%]	ツールによる分析結果（Pair-wise Analysisなど）	評価
データ数量[件 or sec]（追加・削除・拡張）	被覆性	分布[%]	データ数または量[件 or sec]	被覆性	分布[%]	データ数または量[件 or sec]	被覆性	分布[%]	データ数量[件 or sec]	妥当性確認No *,	妥当性確認No, *							正解率[%]	適合率[%]	再現性[%]	F値	その他				

データの妥当性

・・・PoC初期段階　・・・＋PoC最終段階/商品開発開始時　・・・＋商品開発完了時　・・・＋運用時

データ										
妥当性確認No.	出所	選定経緯	時間的妥当性（古すぎなど）	空間的妥当性（場所の特性考慮）	外れ値除去	汚染可能性（改竄含む）	汚染対処方法（Adversarial exampleへの対処含）	検査方法（異常検出方法含む）	ダブルチェックなどの結果	使用可否判断

ラベリング（メタデータ）										総合妥当性判定
妥当性確認No.	出所	選定経緯	処理方法（ラベルポリシー準拠/バラツキ低減）	ラベルの揺らぎ範囲（分散・偏差など）	汚染可能性	汚染対処方法（Adversarial exampleへの対処含）	検査方法（異常検出方法含む）	ダブルチェックなどの結果	使用可否判断	

機械学習モデル・アセスメント票

・・・PoC初期段階　・・・＋PoC最終段階/商品開発開始時　・・・＋商品開発完了時　・・・＋運用時

AI要求分析における機械学習モデルへの要求								AI品質の要求レベル		
過去の実績、POCでの知見	モデル精度（正解率，適合率，再現性，F値，など）				入力特性（空間的・時系列的）	出力特性（多クラス分類，信頼度情報有無，閾値，など）	制約（学習時間，ハイパーパラメータ対象，必要なリソースなど）	外部品質	リスク回避性	AISL**/Lv*
									AIパフォーマンス	AIPL**/Lv*
0	0	0	0	0	0	0	0		公平性	AIFL**/Lv*

MLモデル（ID）	出所（新規・改造・流用，など）	選定経緯	ハイパーパラメータ	ハイパーパラメータ最適化方法
				ブラックボックス最適化（ランダムサーチ，グリッドサーチ，ハイブリッドサーチ，ベイズ最適化法，Nelder-Mead法，遺伝的アルゴリズム方(GA)）
				グレーボックス最適化（データセット・サブサンプリング，学習の早期打ち切り，ウォームスタート(過去の経験ベース)）
				半教師あり学習，模倣学習，逆強化学習
				その他

学習方法（ID）	学習手順	学習終了基準（学習回数・時間・過学習防止方法など）	効率化方法（ツールの利用含む）	敵対的攻撃対策方法

No	機械学習モデルの設計		N回目（No.）機械学習																
	構成（初期値）		学習条件					訓練時											
								正確性（モデル精度）				安定性（ロバスト性）				正確性（モデル精度）			
	MLモデル（ID）	学習方法（ID）	利用するデータセットVer.	利用するMLモデルVer.	訓練用プログラムVer.	妥当性確認用プログラムVer.	検証用（テスト）プログラムVer.	正解率[%]	適合率[%]	再現性[%]	F値	大きな推論外れ	自然界のノイズ	敵対的データ	学習曲線の収束状況（学習時間・学習回数についての見解，学習不足・過学習の判断），ROC曲線/AUC	正解率[%]	適合率[%]	再現性[%]	F値
1																			
2																			
3																			
4																			

妥当性確認時					検証時										仕様の制限（AI以外での処理などへの要求）
安定性（ロバスト性）				実環境の模擬状況	正確性（モデル精度）				安定性（ロバスト性）				実環境との一致性	評価値	
大きな推論外れ	自然界のノイズ	敵対的データ	学習曲線の収束状況（学習時間・学習回数についての見解，学習不足・過学習の判断），ROC曲線		正解率[%]	適合率[%]	再現性[%]	F値	大きな推論外れ	自然界のノイズ	敵対的データ	学習曲線の収束状況（学習時間・学習回数についての見解，学習不足・過学習の判断），ROC曲線			

保守計画アセスメント票

・「計画」と「実績」からなる

■保守計画

・・・PoC初期段階　・・・＋PoC最終段階/商品開発開始時　・・・＋商品開発完了時　・・・＋運用時

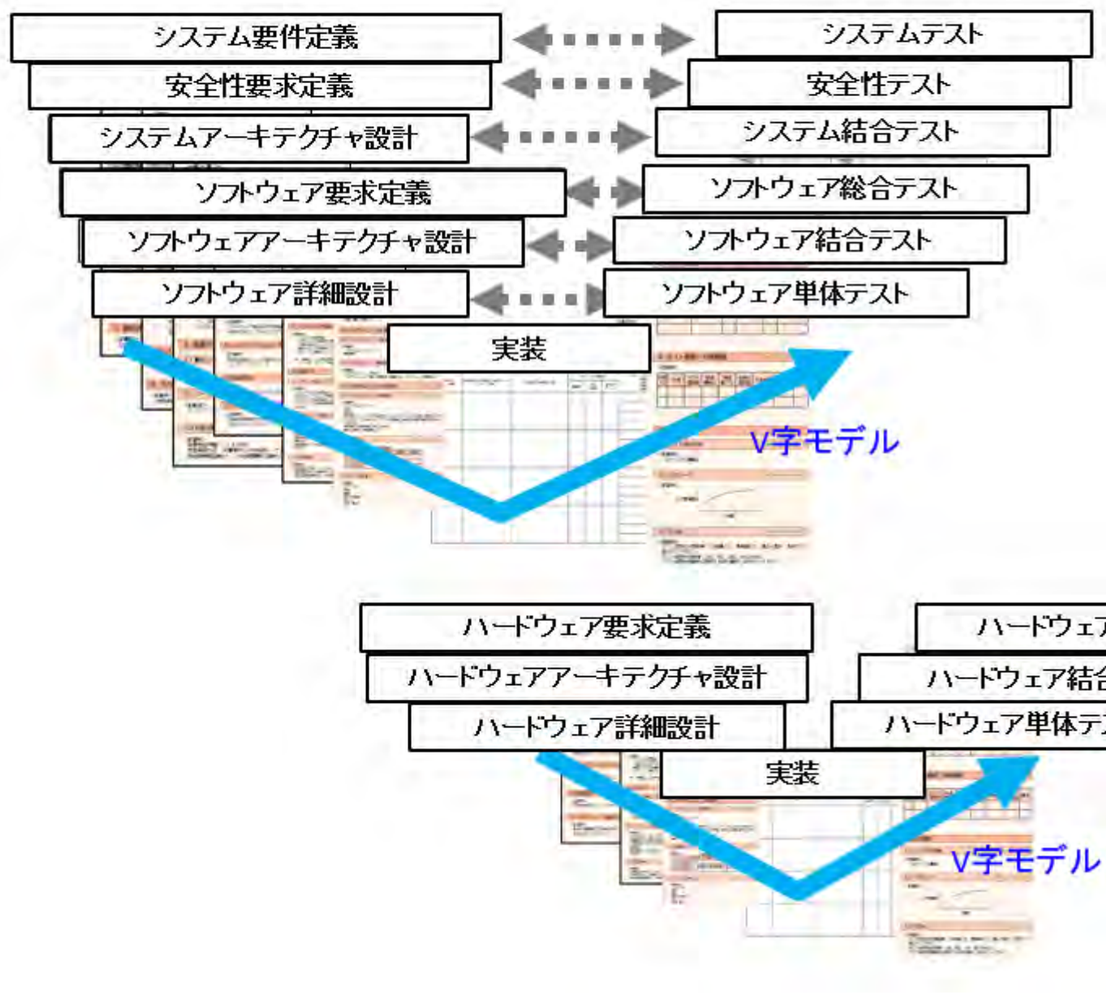
No.	変更対象（What/When）			変更方法(How)	確認方法（How）	変更内容（影響範囲）						
	保守対象	変更の目的 （環境変化への対応/仕様変更・追加・削除/不具合対応/異常への対策）	変更の条件 変化の検出方法（最大・最小・閾値/継続時間） 変更タイミング	変更手段・手順 （再学習，新規学習）	変更後のデグレ防止確認方法	システム要求分析	システムRA	AI要求分析	データセット設計・収集	MLモデル	保全計画	機能安全
1	機械学習モデル	環境変化への対応										
2		仕様変更 （変更・追加・削除）										
3	データセット	環境変化への対応										
4		仕様変更										
5	機能安全 （ソフト・ハード）	環境変化への対応										
6		仕様変更										
7	（人による） 運用方法	環境変化への対応										
8		仕様変更										

■保守実績

Ver.	変更要因	変更内容（影響範囲）							変更結果	
	仕様変更管理票/不具合管理表の登録番号	システム要求分析	システムRA	AI要求分析	データセット設計・収集	MLモデル	保全計画	機能安全	判定（OK/NG）	詳細（他のアセスメントシートへのリンク）
1.00	仕様変更管理表-***	－	－	－	○	○	－	－		
	不具合管理票-***	－	－	－	－	○	－	○		

機能安全プロセス管理

■機能安全プロセス管理の手順と帳票を利用



データセットや学習モデルなどAI要素以外

例）
・データ拡張ツール
・アノテーションツール
・学習モデルを訓練するソフト
・学習モデルをテストするソフト

参考文献

[1] National Institute of Advanced Industrial Science and Technology (AIST), "Machine Learning Quality Management Guideline 2nd Edition," AIST, 2022.

[2] SAE International, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," 15 June 2018. [Online]. Available: https://www.sae.org/standards/content/j3016_201806/.

[3] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 8 April 2018. [Online]. Available: https://arxiv.org/abs/1804.02767.

[4] S. Liu, D. Huang and Y. Wang, "Learning Spatial Fusion for Single-Shot Object Detection," 21 November 2019. [Online]. Available: https://arxiv.org/abs/1911.09516.

[5] A. Bochkovskiy, C.-Y. Wang and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 23 April 2020. [Online]. Available: https://arxiv.org/abs/2004.10934.

[6] G. R. Jocher, "Yolov5," 22 June 2020. [Online]. Available: https://github.com/ultralytics/yolov5.

[7] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," 4 June 2015. [Online]. Available: https://arxiv.org/abs/1506.01497.

[8] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai and H. Ling, "M2Det: A Single-Shot Object Detector based on Multi-Level Feature Pyramid Network," vol. 33, no. 1, pp. 9259-9266, 17 July 2019.

[9] M. Tan, R. Pang , Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," 著: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10778-10787.

[10] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 17 April 2017. [Online]. Available: https://arxiv.org/abs/1704.04861.

[11] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 13 January 2018. [Online]. Available:

https://arxiv.org/abs/1801.04381.

[12] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 4 September 2014. [Online]. Available: https://arxiv.org/abs/1409.1556.

[13] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2016, pp. 2818-2826.

[15] ISO/TC 22/SC 32 Electrical and electronic components and general system aspects, ISO 26262-9:2018 Road vehicles — Functional safety — Part 9: Automotive safety integrity level (ASIL)-oriented and safety-oriented analyses, 2018.

[16] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan and T. Darrell, "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2020, pp. 2633-2642.

[17] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan and T. Darrell, "GitHub Repository of BDD100k," 21 September 2020. [Online]. Available: https://github.com/bdd100k/bdd100k.

[18] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan and O. Beibom, "nuScenes: A multimodal dataset for autonomous driving," 26 March 2019. [Online]. Available: https://arxiv.org/abs/1903.11027.

[19] J. Kim, R. Feldt and S. Yoo, "Guiding Deep Learning System Testing Using Surprise Adequacy," in *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, IEEE, 2019, pp. 1039-1049.

[20] T. Ouyang, V. S. Marco, Y. Isobe, H. Asoh, Y. Oiwa and Y. Seo, "Corner Case Data Description and Detection," in *2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN)*, IEEE, 2021, pp. 19-26.

[21] R. Dabni, "Casting Product Image Data for Quality Inspection, Version 2," 3 July 2020. [Online]. Available: https://www.kaggle.com/ravirajsinh45/real-life-industrial-dataset-of-casting-product.

[22] Y. LeCun and Y. Bengio, "Convolutional Networks for Images, Speech, and Time Series," in *The Handbook of Brain Theory and Neural Networks*, 1995.

[23] H. Taud and J. F. Mas, "Multilayer Perceptron (MLP)," in *Geomatic Approaches for Modeling Land Change Scenarios*, Springer, Cham., pp. 451-455.

[24] T. Mertens, J. Kautz and F. Van Reeth, "Exposure Fusion," in *15th Pacific Conference on Computer Graphics and Applications (PG'07)*, IEEE, 2007, pp. 382-390.

[25] Y. Tian, K. Pei, S. Jana and B. Ray, "Deeptest: Automated Testing of Deep-neural-network-driven Autonomous Cars," in *Proceedings of the 40th International Conference on Software Engineering*, 2018, pp. 303-314.

[26] The Verge, "A Google Self-driving Car Caused a Crash for the First Time," 29 2 2016. [Online]. Available: http://www.theverge.com/2016/2/29/11134344/googleselfdriving-car-crash-report.

[27] T. Ouyang, V. S. Marco, Y. Isobe, H. Asoh, Y. Oiwa and Y. Seo, "Improved Surprise Adequacy Tools for Corner Case Data Description and Detection," *Applied Sciences,* vol. 11, no. 15, 2021.

[28] K. Pei, Y. Cao, J. Yang and S. Jana, "Deepxplore: Automated Whitebox Testing of Deep Learning Systems," in *Proceedings of the 26th Symposium on Operating Systems Principles*, ACM, 2017, pp. 1-18.

[29] Y. Feng, Q. Shi, X. Gao, J. Wan, C. Fang and Z. Chan, "Deepgini: Prioritizing Massive Tests to Enhance the Robustness of Deep Neural Networks," in *Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2020, pp. 177-188.

[30] S. Poulding and R. Feldt, "Generating Controllably Invalid and Atypical Inputs for Robustness Testing," in *2027 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, 2017.

[31] F. Pérez-Cruz, "Kullback-Leibler Divergence Estimation of Continuous Distributions," in *2008 IEEE International Symposium on Information Theory*, IEEE, 2008, pp. 1666-1670.

[32] M. Sokolova, N. Japkowicz and S. Szpakowicz, "Beyond Accuracy, F-score and ROC: a Family of Discriminant Measures for Performance Evaluation," in *Australasian Joint Conference on Artificial Intelligence*, Springer, Berlin, Heidelberg, 2006, pp. 1015-1021.

[33] D. Theckedath and R. R. Sedamkar, "Detecting Affect States Using VGG16, ResNet50 and SE-ResNet50 Networks," *SN Computer Science,* vol. 1, no. 2, pp. 1-7, 2020.

[34] IEEE Standards Coordinating Committee, IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990), IEEE Computer Society, 1990.

[35] J. M. Zhang, M. Harman, L. Ma and Y. Liu, "Machine Learning Testing: Survery, Landscapes and Horizons," *IEEE Transactions on Software Engineering,* 2020.

[36] S. M. Moosavi-Dezfooli, A. Fawzi and P. Frossard, "Deepfool: a Simple and Accurate Method to Fool Deep Neural Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2574-2582.

[37] T. Ouyang, Y. Isobe, V. S. Marco, J. Ogawa, Y. Seo and Y. Oiwa, "AI Robustness Analysis with Consideration of Corner Cases," in *2021 IEEE International Conference on Artificial Intelligence Testing (AITest)*, IEEE, 2021, pp. 29-36.

[38] Y. LeCun, C. Cortes and C. J. Burges, "The MNIST Database of Handwritten Digits," [Online]. Available: http://yann.lecun.com/exdb/mnist/.

[39] ub, "USPS Dataset, Version 1," 7 April 2018. [Online]. Available: https://www.kaggle.com/bistaumanga/usps-dataset.

[40] Blekinge Institute of Technology, "ARDIS: The Swedish Dataset of Historical Handwritten Digits," 2 April 2019. [Online]. Available: https://ardisdataset.github.io/ARDIS/.

[41] Wikipedia, "MNIST Database," [Online]. Available: https://en.wikipedia.org/wiki/MNIST_database.

[42] T. Ouyang, V. S. Marco, Y. Isobe, H. Asoh, Y. Oiwa and Y. Seo, "Corner Case Data Description and Detection," [Online]. Available: https://arxiv.org/pdf/2101.02494.pdf.

[43] T. Byun, S. Vaibhav, V. Abhishek, R. Sanjai and C. Darren, "Input Prioritization for Testing Neural Networks," in *2019 IEEE International Conference On Artificial Intelligence Testing (AITest)*, IEEE, 2019, pp. 63-70.

[44] A. Boopathy, T.-W. Weng, P.-Y. Chen, S. Liu and L. Daniel, "CNN-Cert: An Efficient Framework for Certifying Robustness of Convolutional Neural Networks," [Online]. Available: https://arxiv.org/abs/1811.12395.

[45] T.-W. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, D. Boning, I. S. Dhilon and L. Daniel, "Toward Fast Computation of Certified Robustness for ReLU Networks," [Online]. Available: https://arxiv.org/abs/1804.09699.

[46] L. Ma, F. Zhang, J. Sun, M. Xue, B. Li, F. Juefei-Xu, C. Xie, L. Li, Y. Liu, J. Zhao and Y. Wang, "DeepMutation: Mutation Testing of Deep Learning Systems," in *2018 IEEE 29th International Symposium on Software Reliability Engineering (ISSRE)*,

IEEE, 2018, pp. 100-111.

[47] A. Odena, C. Olsson, D. Andersen and I. Goodfellow, "TensorFuzz: Debugging Neural Networks with Coverage-Guided Fuzzing," *Proceedings of Machine Learning Research,* vol. 97, pp. 4901-4911, 2019.

[48] Kaggle, "House Prices - Advanced Regression Techniques," [Online]. Available: https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data.

[49] United States Census Bureau, "SELECTED HOUSING CHARACTERISTICS," 2020. [Online]. Available: https://data.census.gov/cedsci/table?tid=ACSDP5Y2020.DP04&g=0400000US19.

[50] United States Census Bureau, "American Housing Survey," 2019. [Online]. Available: https://www.census.gov/programs-surveys/ahs/data.html.

[51] A. Lefton and S. Coelho, "This Is the Average Home Size in Every State," [Online]. Available: https://www.bobvila.com/slideshow/this-is-the-average-home-size-in-every-state-53461.

[52] M. Phillip, T. Rusch, K. Hornik and C. Strobl, "Measuring the Stability of Results from Supervised Statistical Learning," *Journal of Computational and Graphical Statistics,* vol. 27, no. 4, pp. 685-700, 2018.

[53] O. Bousquet and A. Elisseeff, "Stability and Generalization," *Journal of Machine Learning Research,* vol. 2, pp. 499-526, March 2002.

編集者・執筆者

国立研究開発法人産業技術総合研究所
デジタルアーキテクチャー研究センター（DigiARC）/
サイバーフィジカルセキュリティ研究センター（CPSEC）/
人工知能研究センター（AIRC）

執筆者一覧

監修: 妹尾義樹
導入章、編集、翻訳: 小西弘一
2 章、3 章: Sumaiya Saima Sultana
6 章、付録 E: 難波孝彰、岡本球夫
7 章、付録 C、付録 D.1:
Sumaiya Saima Sultana, Vicent Sanz Marco, MD Nizam Uddin,
Imrad Zulkar Nyeen
8 章、付録 D.2: Tinghui Ouyang
9 章、付録 D.3: Imrad Zulkar Nyeen, Tinghui Ouyang
10 章、付録 D.4: MD Nizam Uddin, Imrad Zulkar Nyeen
11 章: 藤原清司、難波孝彰
付録 A: Sumaiya Saima Sultana
付録 B: Imrad Zulkar Nyeen, Tinghui Ouyang, Sumaiya Saima Sultana,
MD Nizam Uddin

リファレンスガイドタスクフォースメンバー一覧（2020-21 年度）

Vicent Sanz Marco	産総研
Imrad Zulkar Nyeen	産総研
Tinghui Ouyang	産総研
Sumaiya Saima Sultana	産総研
MD Nizam Uddin	産総研
岡本球夫	パナソニックホールディングス
小西弘一	産総研
妹尾義樹	産総研
中坊嘉宏	産総研
難波孝彰	パナソニックホールディングス
福島真太朗	トヨタ自動車
藤原清司	産総研
三宅和公	住友電気工業
三宅武史	サイバー創研

(敬称略、alphabet/五十音順)

改版履歴

版	日付	記事
1.0	2022 年 7 月 14 日	日本語初版発行。英語 1.2 版を元に以下を加筆。 - 数式番号の追記などの書式変更 - セキュリティ関連の記述をガイドラインへの参照に置換 - 不十分な点を数か所、脚注で指摘