

# 2段階機械学習を用いた危険予測システムアーキテクチャの提案

加地 智也<sup>1</sup> 飯盛 天翔<sup>1</sup> 青山 幹雄<sup>1</sup> 野呂 昌満<sup>1</sup> 沢田 篤史<sup>1</sup>

**概要:** 自動運転や運転支援のソフトウェアとして様々なシステムが提案されている。特に危険予測などでは、交差点など多様な交通状況を想定する必要がある、その状況が持つ複雑性から因果関係を定式化することが稀なので機械学習技術が用いられる。これらの機械学習技術においては、技術の発展に伴い新たな手法が次々と提案されてくる。一方で、危険予測では、実行効率と予測精度のトレードオフが存在する。本研究では、関連技術の進化に追随しながら、実行効率と予測精度を適切に考慮して危険予測システムを開発するための基盤として、2段階の機械学習を用いたソフトウェアアーキテクチャを提案する。危険予測では周辺の状況を検知する1段階目と、検知されたデータから予測を行う2つの機能が必要となり、それぞれに特化した手法があるのでそれらを組み合わせる。提案方法を実際のデータやシミュレーション環境に適用し、予測の精度を確認した。

## A Hazard Anticipation Systems Architecture Using Two-Stage Machine Learning

**Abstract:** A variety of software intensive systems have been proposed for automatic driving and driving support. In particular, hazard prediction functions often employ machine learning based method since these functions are required to adapt various traffic situations. New methods are proposed as technology develops in the relevant area. Moreover, there are trade-off relationships among the required properties of hazard prediction system. For example, prediction accuracy conflicts efficiency. These conflicting requirements often make the system implementation very difficult. Considering the possibility of change, we propose a software architecture that uses two-stage machine learning. Machine learning modules in the first stage of our architecture work to detect traffic circumstances around the car. Modules in the second stage work to estimate the degree of danger from circumstantial properties acquired in the first stage. We have applied the proposed method to actual data and simulation environment, and confirmed the accuracy of the prediction.

### 1. 研究背景と課題

自動運転 [1] や運転支援のソフトウェアとして様々なシステムが提案されている。特に危険予測システムは、物体検知、状態認識、危険度判定など複数の機能から構成される。様々な交通状況に適応するために、それぞれの機能の実現に機械学習技術を導入することが一般化している。センサデバイスの進歩やそれに伴う要求の変化に応じ、新たな機械学習の手法が頻繁に提案される。危険予測においては、実時間性に関する実行効率と予測精度といった対立する要求が存在する。一般的にそれらのトレードオフを考慮しながら実環境で利用可能なシステムを実現することは一

般的に難しい。一方で新たに利用可能となる計測技術やそれに伴って生じる新たな危険予測要求に追随する必要がある。このためには、実行効率と精度という2つの特性に加え、ソフトウェアの変更可能性も考慮しなければならない。本研究では、これら3つの特性の矛盾なく均整の取れた実現をソフトウェア構造設計の問題ととらえ、危険予測のための機械学習モジュールのためのアーキテクチャについて議論する。すなわち本研究では、以下の3つを研究課題とする。

- (1) 危険予測システムに求められる実行効率、予測精度、変更容易性の両立をするための抽象アーキテクチャ設計
- (2) (1)の抽象アーキテクチャに機械学習コンポーネントを内包する具象アーキテクチャの設計
- (3) データセットへの適用による、危険予測システムアー

<sup>1</sup> 南山大学  
Nanzan University, Nagoya, Aichi 466-8673, Japan

キテクチャの有用性確認

## 2. 関連研究

### 2.1 3次元物体検出

3次元物体検出は3次元の物体の位置を特定して認識する技術である。3次元のデータ表現として点群モデルを用いたものがある。これを実現するために2次元の検出器を用いた幾つかのアルゴリズムがある。本研究では3次元の物体を点群ベースで検出する PointPillars[7]の方法を用いて、車両周辺の物体を検出する方法を前提としている。

### 2.2 自動運転システム

竹内 [12]によれば、「安全な自動運転の実現のためには認識技術として、得られた情報を分析する「見えている場所」の認識技術と、地図と自己位置を利用し、情報が得られていない「見えていない場所」の認識技術を併用し、多様な状態を認識する必要がある。」とある。竹内らの論文では、今までは不可視領域だったものを可視領域に拡張して行くことを目的としている。今日の自動運転システムでは、様々なセンサを用いることで外部環境のデータを取得している。主に用いられるものとして、カメラ、ミリ波レーダ、LiDAR、GPS等がある。しかし、これらのセンサを単独で用いても複雑な環境認識をするのは困難である。そのため、これらのセンサを複数用いて認識性能を向上するセンサフュージョン技術が開発されている。主なセンサとそのセンサが取得するデータを1に示す。

| センサ名   | 取得するデータ             |
|--------|---------------------|
| カメラ    | 映像                  |
| ミリ波レーダ | 物体検出とその距離、水平角度、相対速度 |
| LiDAR  | 物体の点群データ            |
| GPS    | 位置情報                |

表 1 主なセンサとその取得データ

#### 2.2.1 認知

自動運転システム [8]では、様々なセンサを用いることで外部環境のデータを取得している。主に用いられるものとして、カメラ、ミリ波レーダ、LiDAR、GPS等がある。これらのセンサを単独で用いても複雑な環境認識をするのは困難である。複数のセンサを用いて認識性能を向上するセンサフュージョン技術が開発されている。

#### 2.2.2 判断

「判断」では、「認知」によって決定した走行可能エリアをもとに自動コマンドシステムが自動車の走る/曲がる/停まるを決定しコマンドを出力する。

#### 2.2.3 制御

判断によって出力されたコマンドによってステアリング、アクセル、ブレーキなどのアクチュエータが「制御」される。

### 2.3 危険予測システムについて

一般的に熟練運転者は、視覚情報だけでなく起こりうる事象の知識や過去に経験に基づいて高度な危険予測と運転駆動の意思決定を行う。橋本ら [5]の論文では、コンピュータが運転責任を負う完全自動運においても、「危険予測したうえで運転行動の意思決定を行うシステム」が必要不可欠であり、このようなシステムを危険予測システムと定義している。竹内 [12]によると自動運転システムは認知、判断、制御からなる。これらから、本研究では、危険予測において行わなければならない処理として、検出と予測の2つがあると考えられる。検出は、車両周囲の物体を認識する処理であり、予測は認知した物体などの状況から危険度を推測する処理である。

### 2.4 自動運転のオープンデータセット

自動運転や深層学習技術の研究用に自動車のビッグデータを持つ様々なデータセットが用意されている。自動車の走行データセットとして、自動車走行中に撮影された車載動画や車載センサからのデータをまとめたものなど、様々な種類のもので用意されている。例えば、オープンデータセットとして Google の兄弟会社の Waymo が作成した Waymo Open Dataset[10] やトヨタの米国法人 Toyota Motor Sales USA とマサチューセッツ工科大学 (MIT) で共同開発した DrivingSeq[3] などがある。本研究では、KITTI Dataset[4]を用いる。KITTI Dataset は、カールスルーエ大学によって作成されたオープンデータセットで、歩行者検出技術や車両検出技術などの研究開発向けに用意されている。データセットの中に LiDAR を用いて取得された 3次元点群データを含み、比較的長時間収集されていて田舎道から高速道路まで幅広く扱っている。KITTI Dataset 内の画像ファイルの一部を図 1 に示す。



図 1 KITTI DATASET の一部

## 3. 研究の進め方

アーキテクチャ設計プロセス並びにその検証過程を図 2 に示す。プロセスは5つのサブプロセスからなり、危険予測システムの全体構造を示す抽象アーキテクチャを定義した後、状況検知部分と危険予測部分の具体化を行う。定義したアーキテクチャに基づくプロトタイプを実装し、データセットの適用と評価を通じて、改善を繰り返すことにより、アーキテクチャの洗練を行う。各サブプロセスの詳細を次の通りである。

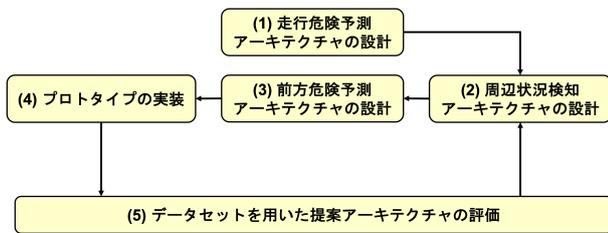


図 2 アーキテクチャ設計プロセス

- (1) 走行危険予測アーキテクチャの設計  
2段階の機械学習モジュール群より構成される抽象アーキテクチャを設計し、インタフェースを決定する。
- (2) 周辺状況検知アーキテクチャの設計  
取得したセンシングデータと機械学習を用いた周辺車両の検出を行うための具象アーキテクチャを設計する。
- (3) 前方危険予測アーキテクチャの設計  
周辺状況検知アーキテクチャを基に機械学習を用いた前方の危険予測を行うための具象アーキテクチャを設計する。
- (4) プロトタイプの実装  
(2), (3) で設計したアーキテクチャに基づいて危険予測システムのプロトタイプを実装する。
- (5) プロトタイプの実験による提案アーキテクチャの評価  
(4) で実装したプロトタイプにデータセットを適用して実験する。その後自動車運転シミュレータに適用し、評価を行う。評価結果から必要に応じて、(2), (3) の設計の見直しを行い、一連のプロセスを繰り返す。

## 4. アーキテクチャ設計

### 4.1 概要

既存の自動運転システムや運転支援システムで用いられているアーキテクチャは図3で示すような構造となっている。本研究ではこのアーキテクチャに基づき必要な危険予測の機能を機械学習モジュールとして埋め込む形で設計する。

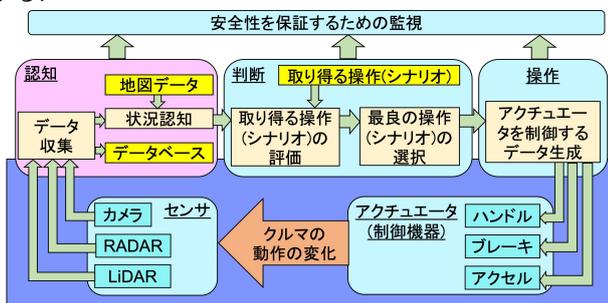


図 3 自動運転システムのモデル

本研究で提案する処理方式を図4に示す。提案する危険予測システムを実現するためには、自車両の周辺状況を検知する機能と検知で得られたデータからこの先危険となり得る事象を予測する機能が必要となる。これらの機能を実現するためのソフトウェア基盤として、本研究では2段階

の機械学習アーキテクチャを提案する。1段階目は、センサーデータから得られる情報から車両周辺の物体を検知する機械学習モジュール群からなる。本研究で実装するプロトタイプシステムでは、PointPillarsを用いて物体検出をし特徴量を抽出する2D CNN (Convolutional Neural Network)を採用する。2段階目は1段階目で抽出する物体の特徴量から危険度を判定する機械学習モジュール群からなる。本研究のプロトタイプでは、1段階目の2D CNNから得られる特徴量をLSTM (Long Short-Term Memory)の入力として用いることで、状況に合わせたリアルタイムな危険予測を実現する。自車両の周辺状況を検知する周辺状況検知と危険予測を行う前方危険予測は異なる機能であるのでその機能に合わせた特性を持つ機械学習を用い、別々のコンポーネントとする。両機能を技術の発展などによって新たな手法に置き換え可能とするためにはインタフェースを決めておけば良い。このように、2段階の機械学習モジュール群からなるアーキテクチャ設計のアプローチをとる。2段階機械学習アーキテクチャはデータフローアーキテクチャに基づいて設計する。これによりLiDARなどで取得される膨大な3次元点群データを効率的に扱えるようにする。

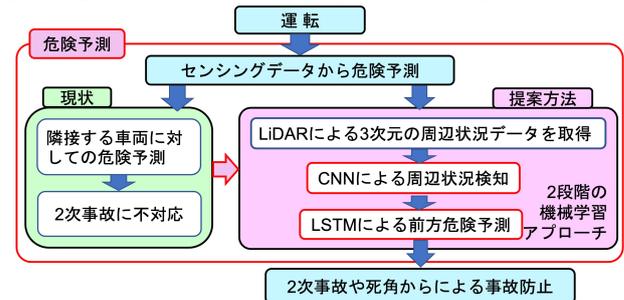


図 4 提案する処理方式

提案アーキテクチャに基づいて構築される危険予測システムはLiDARやカメラなどのセンシングデータを入力とし、走行中の状況で、自車の周辺状況から危険となり得る事象を予測する。危険となる状況はこの先事故が起こる可能性があるかどうかとする。センシングデータからCNNを用いた周辺状況検知と、LSTMを用いた前方危険予測の2段階の機械学習のアプローチをとる。

### 4.2 アーキテクチャ設計指針

本研究では危険予測の機能として予測精度、実行効率に加え、システムの構成要素の変更容易性という目的に対して矛盾なく設計するためのアーキテクチャを提案する。

#### 4.2.1 走行危険予測システムアーキテクチャの設計

本研究の危険予測機能には2.3節で示した通り車両検知と危険予測を行う2つの機能を必要とする。それらは機械学習を用いて実現するので別の機械学習手法への変更の際の容易性や、実行時の精度と効率を考慮する必要がある。この必要とする機能を実現するためのコンポーネントを独立とすることを設計指針とする。以上を踏まえた上で本研究

で提案する抽象アーキテクチャを図5に示す。アーキテクチャでは車両検知と危険予測の機械学習機能を2段階に分割する。このアーキテクチャに基づいて実装される危険予測システムでは、自動車に搭載されたLiDARセンサから収集した3次元点群データを車両検知機能に渡す。LiDARセンサはカメラより広範囲に情報を収集することが可能である。本研究では、LiDARセンサから取得した3次元点群データを高速処理化のために2次元データに変換し物体検出を行う。物体検出から取得されたデータを危険予測機能に渡すことで、危険と予測される車両とそうでない車両を分離することができる。詳細を以下に示す。

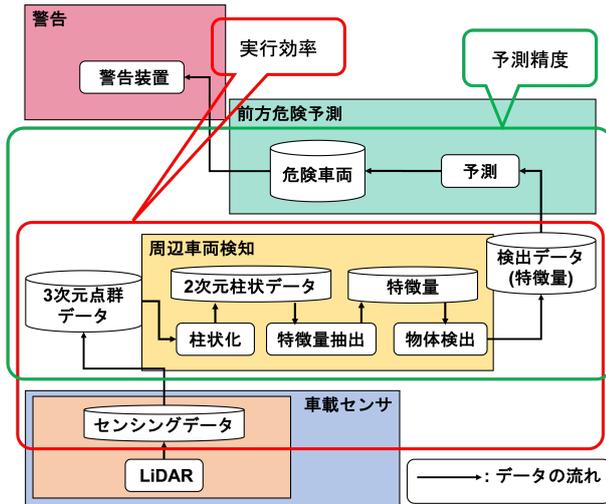


図5 走行危険予測システムの抽象アーキテクチャ

- (1) センサ情報収集  
車載センサのLiDARからセンシングデータとして、3次元点群データを収集する。
- (2) 柱状化  
センサから収集した3次元点群データを2D CNNで処理できるような形に変換する。
- (3) 特徴量の抽出  
(2)で得たデータを2D CNNを用いて特徴量を得る。
- (4) 物体検出  
Single Shot MultiBox Detector(SSD)[13]を用いて自動車などの物体を検出し、同時にそれらの特徴量を抽出する。
- (5) 危険予測器  
物体検出により得られた特徴量をLSTMに適用し、危険であるか、そうでないかを分類する。

#### 4.2.2 周辺状況検知アーキテクチャの設計

周辺状況検知機能では状況検知を行う機械学習を別の機械学習手法に容易に変更すること、実行時に素早く高精度に検知を行う必要がある。この要求を満たすために、センサからセンシングデータを収集するコンポーネントとそのデータを用いて周辺状況を検知するコンポーネントに独立させて設計した。これらを踏まえた上で周辺状況検知機

能の具象アーキテクチャを図6に示す。この機能では車載LiDARセンサから取得した3次元点群データを高速処理するために、3次元から2次元へと変換し、2次元データを入力に使い、物体検出を行ったのち、検出した物体をクラスごと(自動車, 歩行者, サイクリスト等)に分類し、それぞれの特徴量を取得する。

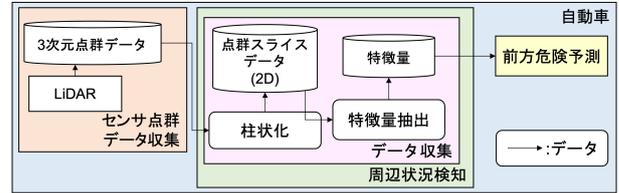


図6 周辺状況検知のための具象アーキテクチャ

#### 4.2.3 前方危険予測アーキテクチャの設計

機械学習技術において、技術の発展に伴い新たな手法が次々と提案されてくるので危険予測機能では周辺状況検知機能同様に予測を行う機械学習を別の機械学習手法へ容易に変更すること、セーフティクリティカルなシステムであるから実行時に実時間性を持って高精度の予測を行う必要がある。一般的な機械学習の枠組みにのっとり、特徴量を抽出するコンポーネント、危険を予測するコンポーネントを独立させて設計した。これを踏まえた上で前方の危険予測機能の具象アーキテクチャを図7に示す。ここでは、車載のLiDARセンサの利用をた設計を示している。自動車に搭載されているLiDARセンサから取得できる3次元点群データを基に物体検出を行う。前方状況検知機能から得られた特徴量を用いて、危険と予測される車両とそうでない車両を分離する。

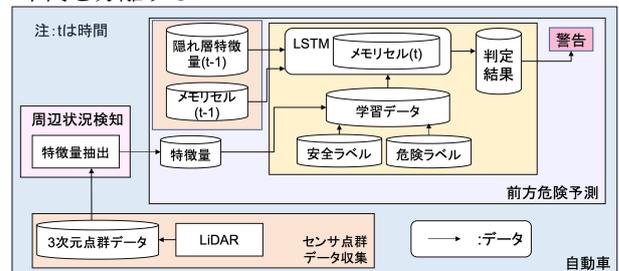


図7 前方危険予測のための具象アーキテクチャ

- (1) 特徴量の取得  
周辺状況検知機能によって抽出された特徴量を取得する。
- (2) 学習データの生成  
特徴量を危険であるものとそうでないものでラベル付けを行い、それらを学習データとして生成する。
- (3) LSTMの適用  
学習データおよび時間t以前の隠れ層の特徴量とメモリセルのデータをLSTMへ適用し、時間tにおけるメモリセルに保存する。
- (4) 危険予測  
LSTMを用いて、取得したデータから危険であるかそ

うでないかの判定を行う。

(5) 警告システムへの出力

予測された判定結果を警告システムへデータとして出力する。

## 5. プロトタイプの実装

### 5.1 実装の目的

プロトタイプ実装の目的は提案するアーキテクチャの妥当性評価である。主に次の4つの観点からアーキテクチャが妥当であるかを評価する。

(1) 提案アーキテクチャの妥当性確認

提案アーキテクチャの動作が妥当であるか確認する。

(2) 3D CNN を用いた検出及び予測と比較し用いた検出, 提案アーキテクチャの優位性を確認

機能を分割せずに単体の3DCNNを用いた検出システム及び予測と比較し, 提案アーキテクチャを使用した際の優位性を確認する。

(3) 危険と予測される車両の確認

車両検出結果から危険と予測される車両とそうでない車両を区別できるか確認する。

(4) シミュレータ適用に向けた確認

シミュレータ上でシステムを動かすためにプロトタイプで結果を分類できるか確認する。

### 5.2 実装環境

本研究でのプロトタイプを作成し実験するために必要な実装環境及び機械学習の機能の実現のために必要なコンポーネントを表2と表3に示す。

| システム  | システム名                     |
|-------|---------------------------|
| OS    | Ubuntu 18.04.5 LTS        |
| プロセッサ | Intel Core i7-8700K       |
| メモリ   | DDR4-2666 32GB            |
| GPU   | NVIDIA GeForce GTX 1080Ti |

表2 ハードウェア環境

| コンポーネント名     | バージョン  |
|--------------|--------|
| Python       | 3.6.9  |
| NumPy        | 1.18.5 |
| scikit-learn | 0.19.1 |
| Pandas       | 1.1.1  |
| TensorFlow   | 2.3.1  |
| Matplotlib   | 3.3.1  |
| Keras        | 1.1.2  |

表3 ソフトウェア環境

実装言語としてPythonを用いる。NumPy, Pandasを用いて学習データの整形などを行いTensorFlow, Kerasを用いて機械学習コンポーネントを実装する。Matplotlibにより, 研究結果のグラフを作成する。

## 6. 実験結果と評価

### 6.1 KITTI データセット適用の目的

本研究で提案したアーキテクチャを基に実装したプロトタイプを, 次の2つのデータに対して適用し, 妥当性と有効性の評価を行う。

(1) 事故が起こりそうな状況が含まれるデータ

(2) 安全な状況のデータ

提案方法により作成されたシステムがどれくらいの精度で危険を予測できるのか, 提案されてきたシステムと比較してどれくらいの精度となっているのかを検証する必要があるため, 本研究のプロトタイプを用いた検証を行う。2つのデータに対して, 本研究が提案する方法で危険であるかまたは安全であるかを分類できるかという観点で妥当性を評価する。

LiDAR センサを使用している本研究の提案方法と, カメラを用いた危険予測と比較し有効性を評価する。

### 6.2 適用対象データ

走行データとしてKITTI Datasetの中から走行シーンのLiDARデータ約7,500個を用いた。プロトタイプの物体検出及び危険予測の実験においてKITTI Datasetを用いる。KITTI Dataset内の複数のシーンから周囲に車両が存在しない場合, 車間距離が十分にあると見える場合などを安全と思われる状況として用いる。また, 周囲に多数車両が存在する場合, 渋滞中の状態や合流する車両がある場合などを危険と思われる状況とし, 以下に危険と思われる状況を13個のファイル, 安全と思われる状況を1個のファイルとして実験に用いた。

### 6.3 プロトタイプへの適用プロセス及び構成

本研究の適用プロセス及びプロトタイプの構成を図8に示す。

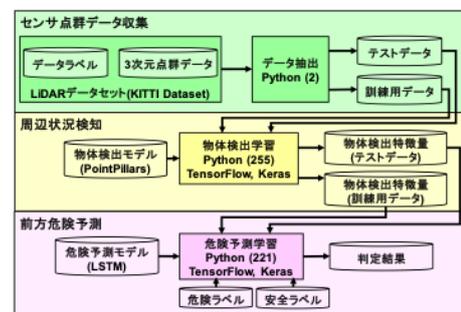


図8 プロトタイプの構成

(1) 人手によるデータ分割

本研究で取り扱うデータは, 複数の運転場面が順番に一つのデータにまとまっている。本研究では, 場面ごとにデータを分類してから利用する。その概要を図9に示す。

(2) PointPillars による物体検出及び特徴量抽出

本研究では、(1)で整形したデータを基に PointPillars を用いて物体検出を行うことで特徴量を抽出することができる。

(3) Python によるデータ整形

(2)で得た特徴量のデータを危険予測の学習に使用するためのデータを作成する。a. 特徴量をそれぞれの場面において分割して出力しているため複数のデータをそれぞれ読み込む。b. 危険が含まれている場面のデータと、安全な状況のデータの二つに整形する。

(4) Numpy によるラベル付け

(3)で二つに整形されたデータに対して、危険というラベルと安全というラベルでラベル付けを行う。

(5) Python による学習データとテストデータの作成

(4)で整形したデータの割合を指定し学習データとテストデータに分割する。

(6) 前方危険予測

(5)で作成した学習データを入力として LSTM に適用し学習させる。また、学習したモデルに対しテストデータを入力し結果を得る。

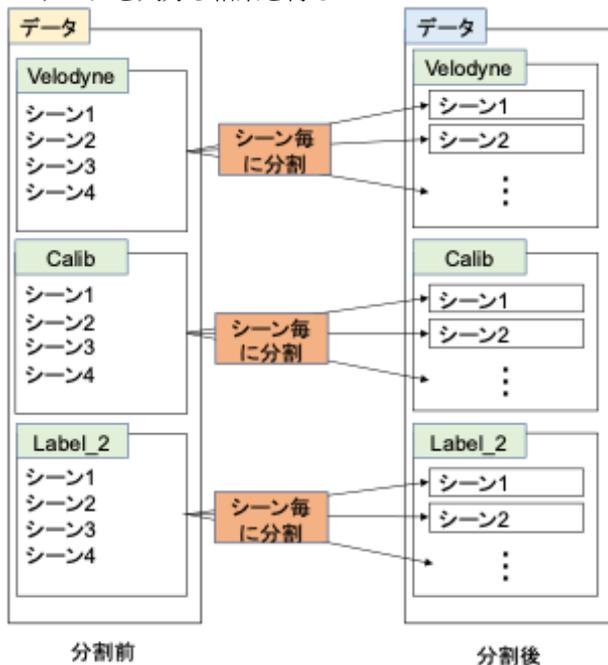


図 9 データ分割の概要

6.4 プロトタイプへの適用結果

各データに対しプロトタイプを適用した結果を表 4, 表 5, 図 10, 図 11 はプロトタイプにデータセットを適用し物体検出及び危険予測を実験した際の結果である, 表 4 はデータセット全体に対しての物体検出の学習時及び特徴量抽出の実行時間の結果, 表 5 は全てのデータセットに対し前方危険予測にかかった時間, 図 10 と図 11 は前方危険予測のテスト結果である。

| 実行内容         | 実行時間  |
|--------------|-------|
| 物体検出         | 2.50h |
| 物体検出 (特徴量抽出) | 0.15h |

表 4 物体検出の実行時間

| 実行内容 | 実行時間 |
|------|------|
| 危険予測 | 251s |

表 5 前方危険予測の実行時間

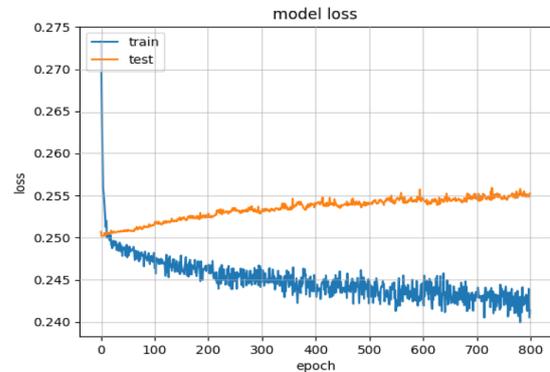


図 10 適用結果 (loss)

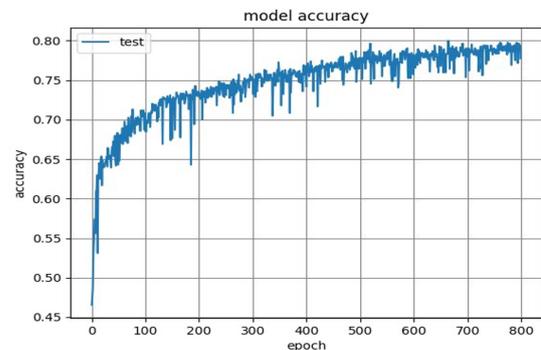


図 11 適用結果 (accuracy)

6.5 2段階の機械学習アーキテクチャの評価

本研究では、周辺状況検知と前方危険予測に分けてそれぞれ 2D CNN と LSTM の 2 段階機械学習手法を用いたアーキテクチャを提案した。周辺状況検知と前方危険予測が異なる機能であり、その機能に特化して独立して進化する機械学習手法に対応可能とするためである。これによって周辺状況検知と前方危険予測が独立して設計でき、かつ、機械学習も独立して行うことができるという利点がある。本研究で提案したアーキテクチャでは、LiDAR センサを用いることを前提としており、この LiDAR センサから得られる 3 次元点群データはデータ量が非常に大きく、処理の負荷がかかるものである。このことより、データ処理の観点から物体検出と危険予測の機能を分割し、より適したアルゴリズムを用いることによって処理を円滑に行えることが確認できた。

## 6.6 データ数と時間の評価

図 12 に示すように、事前に学習をしたモデルを用いて危険予測を行うことで様々な運転状況において 0.25 秒以下で危険予測を行うことが確認できた。また、物体検出については 30 フレームを一回のセットとしてテストしていたので 1 フレームあたり 0.3 秒で行い、危険予測については 1 フレームあたり 0.0002 秒で行うので物体検出後のデータを危険予測にかけた場合でも少ない時間で実行できることが確認できた。

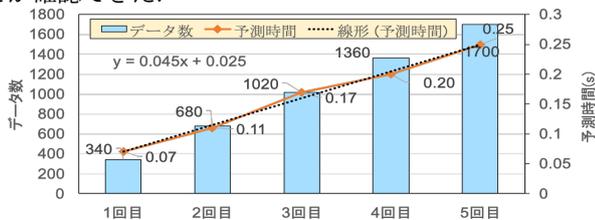


図 12 予測の計算時間

## 6.7 プロトタイプの正答率の評価

プロトタイプの学習データの変化による損失と正答率を表 6 に示し、その結果のグラフを図 13 に示す。TensorFlow を用いてプロトタイプを実装し、KITTI Dataset の LiDAR データに適用して評価を行った。周辺状況検知は、学習に約 2.5 時間、テストは 1 回につき 30 セットの LiDAR データに対して約 10 秒となった。6.4 節の結果で示した通り、危険予測での学習の結果は正答率が約 75% という結果が確認できた。多くの場合で安全と危険の予測が可能で、テストでは少ない時間で結果を確認できた。テスト用データを複数パターン用意して評価した場合も正答率が 70%~80% となる結果が確認できた。

| データ数 (枚) | 損失 (loss) | 正答率 (accuracy) |
|----------|-----------|----------------|
| 340      | 0.243     | 0.736          |
| 680      | 0.241     | 0.798          |
| 1020     | 0.242     | 0.763          |
| 1360     | 0.247     | 0.745          |
| 1700     | 0.245     | 0.749          |

表 6 データ数と損失と正答率の内訳

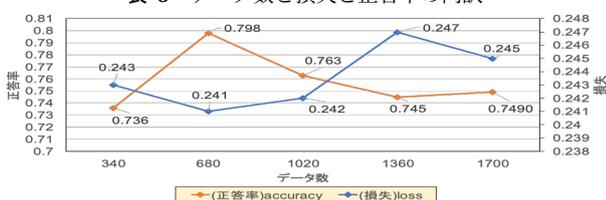


図 13 データ数による損失と正答率

## 7. 考察

### 7.1 先行研究の結果と本研究の比較

先行研究 [9] の結果では、Subject ID を使用している。実験結果では使用しているモデルやデータによって異なるが、ほとんどの Subject ID では正答率が 80% 以上となっている。

これに対し、本研究のプロトタイプの結果では正答率が 70~80% となっている。これは先行研究と比較して正答率が低い結果である。これは学習データ不足による学習不足が原因だと考える。特徴量のデータ数が 3,119 枚 (安全データ 1,505 枚, 危険データ 1,614 枚) のときの学習では正答率がおよそ 50% となっていたが、特徴量のデータ数が 3,396 枚 (安全データ 1,505 枚, 危険データ 1,891 枚) に増やしたときの学習では正答率がおよそ 70% という結果となっているからである。

### 7.2 評価に基づく提案アーキテクチャの考察

本研究で提案したアーキテクチャは、現在主流となっているカメラなどを使用した運転支援システムがあり、それでは遠方の自動車の把握やそれに伴った危険予測や二次事故の防止が困難である。この課題に対しアーキテクチャを提案しプロトタイプを作成した。先行研究 [6] では、カメラからの 2D 映像を基に LSTM を用いた新しい事故検知アルゴリズムについて優位であるとしている。これに対し、本研究では、物体検出を 2D CNN で行い、LSTM を用いて危険予測を行い正答率を測定した。この実行結果より約 70~80% の正答率を確認した。以上のことより本研究の提案アーキテクチャは 3D データという 2D より膨大なデータに対して危険予測を 2 段階に分けることによって実行効率を損なわず同等の予測精度及び変更容易性を持ったアーキテクチャを提案できた。つまり、提案アーキテクチャを基にしたプロトタイプの実験結果からカメラの 2D データを用いた LSTM 単体による事故検知アルゴリズムと同等の特性を持ったアーキテクチャを提案できた。

### 7.3 提案アーキテクチャのコンポーネントの考察

本研究の提案方法では、前方に対しての危険予測を行っていたが学習データによって前方だけでなく後方や横などの自車の周囲の危険を予測が可能になる。また、周辺状況検知システムや前方危険予測システムをコンポーネントとして組み替えることも可能で、より良い機械学習の手法やアルゴリズムが提案された時に変更することができる。これらのことからアーキテクチャとしてコンポーネントの変更可能性があると考えられる。

## 8. 検証上の問題と今後の課題

検証上の問題は以下の 3 点である。

### (1) プロトタイプ及び提案アーキテクチャの再検証

本研究のプロトタイプを用いた検証では、機械学習を用いた危険予測の正答率がセーフティクリティカルなシステムとしては十分な結果が得られなかった。考察で述べた通り、これは学習不足の可能性があり、まだ検証しきれていないとは言えない。それゆえ、学習データを増やすなど本研究のプロトタイプ及びアーキテク

チャを検証し直す。組み込みシステムとして CPU やメモリなどハードウェアの制約はアーキテクチャの機能の変更容易性を犠牲にする可能性があるので議論の上、提案方法の評価をする。

### (2) 機械学習によるデータセットの分類

本研究では学習用データセットの分類を手で行ったので主観が入っている部分がある。今後はこのデータセットの分類にも機械学習を用いることで主観を取り払って実験を行う。

### (3) 別データ適用時の有用性の評価

本研究では、データセットとして KITTI Dataset を使用して危険予測を行った。Waymo Open Dataset 等の他のデータセットをプロトタイプに適用し、危険予測を行った際の適用結果を比較したのち、有用性を確認する必要がある。

今後の課題は以下の 2 点である。

### (1) アスペクト指向に基づくアーキテクチャ設計の検討

本研究の提案アーキテクチャを実行効率と予測精度に関する視点からアスペクト指向アーキテクチャに定義することについて検討する。

### (2) シミュレータにおける走行実験

提案した危険予測システムアーキテクチャをシミュレータ [2][11] に適用し、走行実験を行い、システム全体の動作を確認する必要がある。

## 9. まとめ

自動車に高度な運転支援システムが搭載されている中、機械学習を用いたシステムによって危険を予測することの実現は困難である。そこで本研究では、点群データを 2DCNN を用いた周辺状況検知システムアーキテクチャと LSTM を用いた前方危険予測システムアーキテクチャによる 2 段階の機械学習を用いた危険予測システムアーキテクチャを提案した。提案したアーキテクチャのプロトタイプを TensorFlow と Keras によって実装し、KITTI Dataset をそのプロトタイプに適用し評価した。先行研究との比較から、提案アーキテクチャは同等の結果が得られることが確認できた。3 つの検証上の問題と 2 つの今後課題を次の研究で行う。本研究で提案した 2 段階の機械学習の機能を持つアーキテクチャの提案は、著者が知る限り提案されておらず、今後の機械学習を用いた車載ソフトウェアのアーキテクチャの応用に意義があると考えられる。

## 参考文献

- [1] 青木 啓二, 自動運転車の開発動向と技術問題, 情報管理, 2017, Vol. 60, pp. 229-239.
- [2] A. Dosovitskiy, et al., CARLA: An Open Urban Driving Simulator, Proc. of CoRL 2017, PMLR Vol. 78, pp. 1-16, 2017. 入手先 (<http://proceedings.mlr.press/v78/>.)
- [3] L. Ding, et al., MIT DriveSeg (Manual) Dataset for Dynamic Driving Scene Segmentation. Massachusetts Institute of Technology AgeLab Technical Report 2020-1, 2020.
- [4] A. Geiger, et al., Vision Meets Robotics: The KITTI Dataset, International Journal of Robotics Research, Vol. 32, No. 11, pp. 1231-1237, 2013.
- [5] 橋本 康平, 他, 論理知識型 AI に基づく自動運転のための危険予測システムの構築と評価, 第 61 回制御システム情報学会研究発表講演会, Vol. 31, No. 5, pp. 191-201, 2018.
- [6] D.-S. Kim, et al., Traffic Accident Detection Based on Ego Motion and Object Tracking, Journal of Advanced Information Technology and Convergence, Vol. 10, No. 1, pp. 15-23, 2020.
- [7] A. H. Lang, et al., PointPillars: Fast Encoders for Object Detection from Point Clouds, Proc. of CVPR 2019, IEEE, pp. 12697-12705, 2019.
- [8] 三浦 遥夏, 他, 自動運転システムアーキテクチャに基づいた走行可能エリアの決定, 第 61 回自動制御連合講演会論文集, pp. 845-850, 2018.
- [9] S. Mozaffari, et al., Deep Learning-Based Vehicle Behavior Prediction for Autonomous Driving Applications: A Review, IEEE Trans. ITS, Vol. 23, No. 1, pp. 33-47, 2022.
- [10] P. Sun, et al., Scalability in Perception for Autonomous Driving: Waymo Open Dataset, Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2446-2454, 2020.
- [11] S. Shah, et al., AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles, Field and Service Robotics, Proc. of FSR 2017, 2017.
- [12] 竹内 栄二郎, 自動運転システムにおける情報処理技術の最新動向: 2. 環境認識 (認知) 技術, 情報処理, Vol. 57, No. 5, pp. 441-445, 2016.
- [13] Liu W. et al., SSD: Single Shot MultiBox Detector, Computer Vision — ECCV 2016, Leibe B., Matas J., Sebe N., Welling M. (eds), Lecture Notes in Computer Science, Vol. 9905. Springer, 2016.