

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の設計方法の提案と評価

大鷹 弘史[†] 加地 智也[†] 飯盛 天翔[†] 吉本 凌也[†] 青山 幹雄[†]

南山大学 理工学部 ソフトウェア工学科[†]

1 研究の目的と課題

本稿では、道路上での自転車に対する危険を予測しドライバーへの事前警告や回避行動可能な2段階の機械学習を用いた危険予測システムのアーキテクチャ開発方法を提案し、LiDAR データに用いて評価を行う。本稿では上記の目的のために、次の二つを研究課題とする。

- (1) センシングデータと機械学習を用いた危険予測システムアーキテクチャの設計方法の提案
- (2) 自動車走行シミュレータに適用し、危険予測システムアーキテクチャの有用性の評価

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。本稿では3次元の物体を点群ベースで検出するPointPillars[3]の方法を用いる。

2.2 機械学習を用いた危険予測技術[1, 4]

機械学習を用いた危険予測技術としてベイジアンネットワークやCNNなどを用いた方法の提案がある。

2.3 自動運転のオープンデータセット[2, 5]

自動運転やディープラーニングの研究用に自動車のビッグデータを持つデータセットが用意されている。

3 アプローチ

本稿で提案するアプローチを図1に示す。

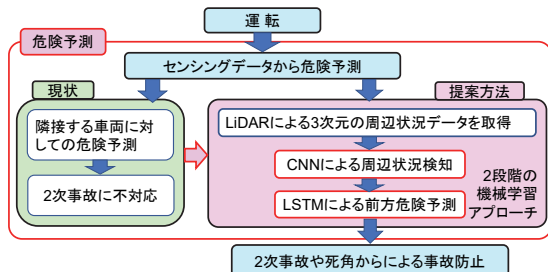


図1 アプローチ

本稿で提案す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、直進走行と

交差点の二つの状況で、自車の周辺状況から危険となり得る事象を予測する。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 提案方法

4.1 アーキテクチャ設計プロセス

5つのステップから成る設計プロセスを提案する。

- (1) 走行危険予測アーキテクチャの設計
自動車走行データを基に自動車走行中に危険予測を行うアーキテクチャを設計する。
- (2) 周辺状況検知アーキテクチャの設計
3次元点群データから周辺車両の検出を行う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前方の危険予測を行う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を実装する。
- (5) シミュレータを用いた提案アーキテクチャの評価
(4)で実装したプロトタイプを自動車運転シミュレータに適用し、評価を行ったのち、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。

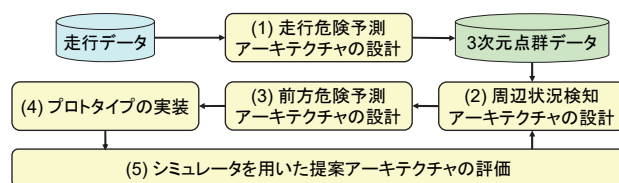


図2 アーキテクチャ設計プロセス

4.2 周辺状況検知アーキテクチャの設計

周辺状況検知アーキテクチャの構成を図3に示す。

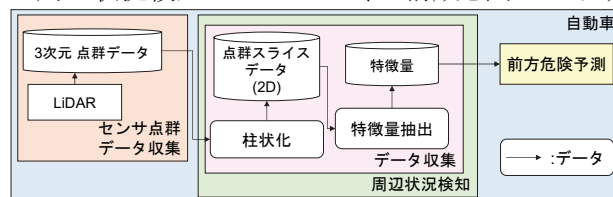


図3 周辺状況検知アーキテクチャ

各コンポーネントの機能を以下に示す。

- (1) 点群データの取得

A Design Methodology of Hazard Anticipation Systems Architecture Using Machine Learning and its Evaluation
[†]Hirofumi Ohtaka, Tomoya Kachi, Tensho Iimori, Ryoya Yoshimoto, Mikio Aoyama
[†]Department of Software Engineering, Nanzan University

- LiDAR から車両前方の 3 次元点群データを得る。
 (2) 点群データの柱状化
 3 次元点群データを PointPillars 内の Pillar Feature Net に通して 2 次元データ(Pillar)に変換する。
 (3) 特徴量の抽出
 2 次元データを CNN へ適用し、特徴量を抽出する。

4.3 前方危険予測アーキテクチャの設計

本稿では運転時の前方と交差点での危険予測を行うため、物体認識の結果を時系列データに変換し、LSTM を用いた前方危険予測の方法を提案する。提案方法を用いた前方危険予測アーキテクチャの構成を図 4 に示す。

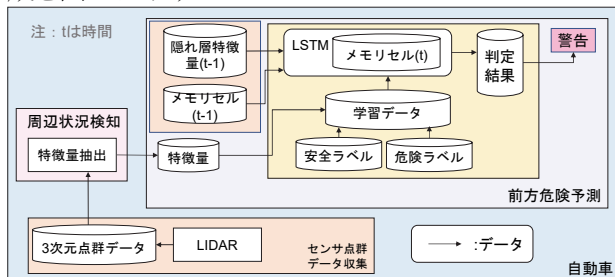


図 4 前方危険予測アーキテクチャ

5 プロトタイプの実装

5.1 プロトタイプの実行環境

プロトタイプ実装環境を表 1, 表 2 に示す。

表 1 ハードウェアコンポーネント

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 コア数 3,584

表 2 ソフトウェアコンポーネント

コンポーネント名	バージョン
TensorFlow	2.3.1
Keras	1.1.2
Python	3.6.9

5.2 プロトタイプの構成

プロトタイプの構成を図 5 に示す。

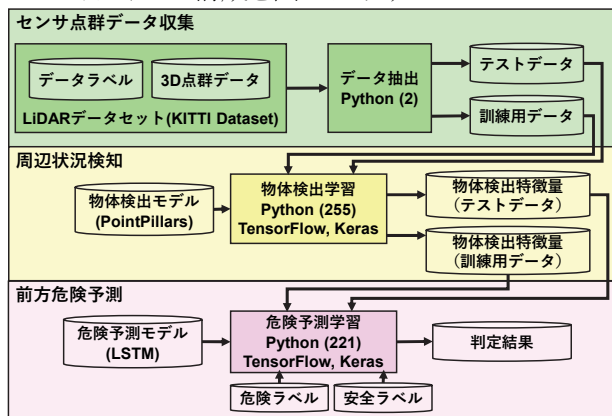


図 5 プロトタイプの構成

6 実行結果

- (1) 走行データとして KITTI Dataset[2]から図 6

に示す走行シーンの LiDAR データ 7,500 個を用いた。



図 6 実行対象の走行シーン

- (2) 周辺状況検知の結果として得られた特徴量の一部を図 7 に示す。この特徴量を NumPy の reshape を用いて 2 次元(1,400 行×9,072 列)に変換して前方危険予測のために LSTM に渡す。

```

379 -0.21825004 -0.09568507 -0.0295704 -0.1546157 -0.0624421 0.1177142
380 -0.21825004 -0.09568507 -0.0295704 -0.1546157 -0.20279641 -0.07323599
381 -0.21825004 -0.04994417 0.45762092 -0.1546157 -0.29778063 -0.12116629
382 -0.21825004 0.06171964 0.24548253 -0.03267828 0.49542555 0.05131567
383 -0.20089298 -0.05858648 0.48315772 -0.1546157 -0.5481606 -0.0799323
384 -0.21242519 -0.09568507 -0.0295704 -0.1546157 -0.02133786 -0.13041696
385
386 -0.21825004 -0.09568507 -0.0295704 -0.1546157 -0.02133786 -0.13041696
387 -0.21825004 -0.09568507 -0.0295704 -0.1546157 -0.02133786 -0.13041696
388 -0.21825004 -0.04994417 0.45762092 -0.1546157 -0.02133786 -0.13041696
389 -0.21825004 0.06171964 0.24548253 -0.1546157 -0.02133786 -0.13041696
390 -0.20089298 -0.05858648 0.48315772 -0.1546157 -0.02133786 -0.13041696
391 -0.21242519 -0.09568507 -0.0295704 -0.1546157 -0.02133786 -0.13041696
392
393 -0.21825004 -0.09568507 -0.0295704 -0.1546157 0.5692138 -0.2169511
394 -0.21825004 -0.09568507 0.0295704 -0.1546157 0.0479089 -0.14607225
395 -0.21825004 -0.04994417 0.45762092 -0.1546157 -0.0624421 -0.2169511
396 -0.21825004 0.06171964 0.24548253 -0.1546157 -0.0624421 -0.07232332
397 -0.20089298 -0.05858648 0.48315772 -0.1546157 -0.0624421 -0.2169511
398 -0.21242519 -0.09568507 -0.0295704 -0.1546157 -0.02133786 -0.13041696
    
```

図 7 2次元特徴量(一部)

7 評価

- (1) 2 段階の機械学習アーキテクチャの設計評価

本稿では、周辺状況検知と前方危険予測に分けてそれぞれ CNN と LSTM を用いたアーキテクチャを提案した。これは周辺状況検知と前方危険予測が異なる機械学習を行う必要があるためである。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。

- (2) プロトタイプの評価

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。危険予測の結果については別途報告する。

8 まとめ

本稿では LiDAR の 3 次元点群データから 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 段階の機械学習システムアーキテクチャの設計方法を提案した。設計したアーキテクチャのプロトタイプを TensorFlow と Keras を用いて実装した。KITTI Dataset の LiDAR データに適用して、提案アーキテクチャの評価を行った。

参考文献

- [1] F. Althé, et al., An LSTM Network for Highway Trajectory Prediction, Proc. of ITSC 2017, IEEE, Oct. 2017, pp. 353-359.
- [2] A. Geiger, et al., Vision Meets Robotics: The KITTI Dataset, Int'l J. of Robotics Research, Vol. 32, No. 11, Sep. 2013, pp. 1231-1237.
- [3] A. H. Lang, et al., PointPillars: Fast Encoders for Object Detection from Point Clouds, Proc. of CVPR 2019, IEEE, Jun. 2019, pp. 12697-12705.
- [4] S. Mozaffari, et al., Deep Learning-Based Vehicle Behavior Prediction for Autonomous Driving Applications: A Review, IEEE Trans. ITS, Early Access, Aug. 2020, pp. 1-15.
- [5] M. M. Trivedi, et al., Attention Monitoring and Hazard Assessment with Bio-Sensing and Vision: Empirical Analysis Utilizing CNNs on the KITTI Dataset, Proc. of IV 2019, IEEE, Jun. 2019, pp. 1673-1678.