

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設計方法の提案と評価

2017SE019 飯盛 天翔 2017SE027 加地 智也 2017SE065 大鷹 弘史 2017SE112 吉本 凌也
指導教員 青山 幹雄

1 研究背景と課題

1.1 研究背景

本稿では、道路上での自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 2 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設計方法の提案
- (2)シミュレータおよびデータセットに適用し、危険予測システムアーキテクチャ設計方法の有用性確認

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 自動運転システム

竹内[5]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。

2.2.1 モデル

自動運転システムのモデルを図1に示す。自動運転システムは、主に外部環境を取り込む「認知」、その外部環境から取り込んだデータを処理して走行方法を決定する「判断」、その決定したデータをもとに外部環境に影響を与えるアクチュエータを制御する「操作」からなる。

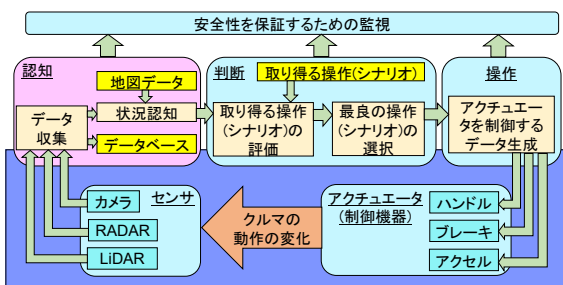


図1 自動運転システムのモデル

2.2.2 認知[1]

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表1に示す。

表1 主なセンサとその取得データ

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

自動運転中、外部環境、ドライバ、自車両の状態、走行目的、手動操作を認知システムが分析して走行可能エリアを決定する。

2.3 自動運転のオープンデータセット[2]

自動運転や Deep Learning 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。本稿で用いる自動車の走行データセットは、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であり、様々な種類が用意されている。

3 アプローチ

3.1 前提条件

本稿の前提条件を示す。

- (1) LiDAR センサの搭載

前方自動車の検出には、自車に3次元点群データを収集できるLiDARを搭載する。

3.2 アプローチ

本稿で提案するアプローチを図2に示す。提案する危険予測システムの実現は、自車両の周囲状況からこの先危険となり得る事象を予測し、その事象をドライバーに警告することが必要となる。それらのセンシングデータから得た情報を基に PointPillars で物体検出をし、その特徴量を LSTM 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ことができるのではないかと考える。周辺状況検知アーキテクチャと前方危険予測アーキテクチャでは、異なる機能、異なるアルゴリズムであるため別々のコンポーネントにしておき、インターフェースのみ決めておく必要がある。そのため、本稿では 2 段階の機械学習アプローチをとる。また、LiDAR で取得される 3 次元点群データは膨大であるた

め、本稿ではデータフローアーキテクチャとする。

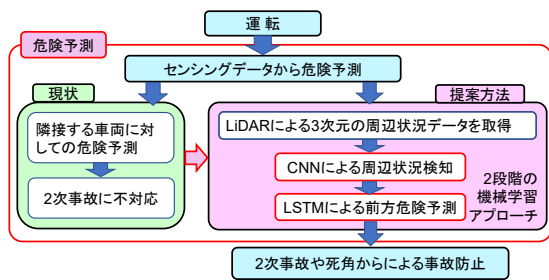


図2 アプローチ

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。

設計プロセスの図3に示す。

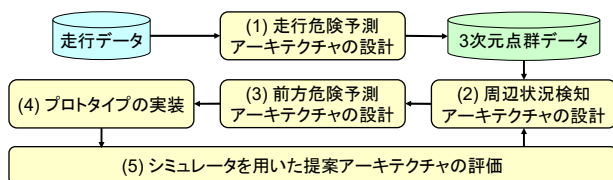


図3 アーキテクチャ設計プロセス

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4に示す。

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。

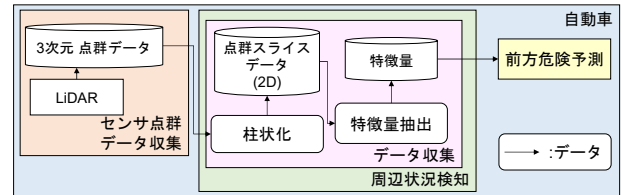


図4 周辺状況検知アーキテクチャ

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

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、周辺状況検知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5に示す。

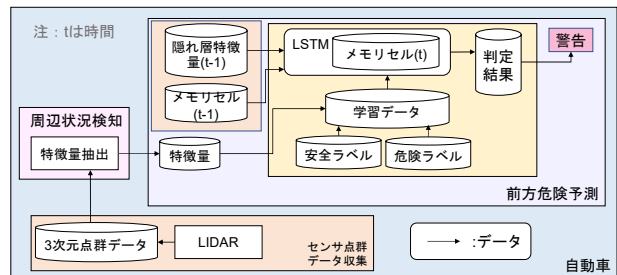


図5 前方危険予測アーキテクチャ

詳細を以下に示す。

(1) 特徴量の取得

周辺状況検知アーキテクチャ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かそうでないか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 内の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 プロトタイプの実行環境

プロトタイプ実装環境を表 2, 表 3 に示す。

表 2 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 3 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 6, 7 に示す。

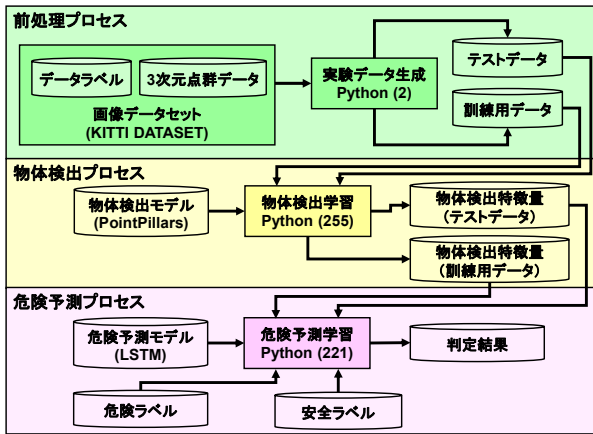


図 6 プロトタイプの構成

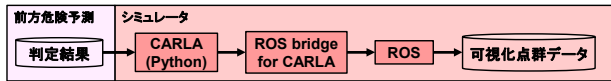


図 7 シミュレータの構成

6 KITTI Dataset への適用結果

- (1) 走行データとして KITTI Dataset[2]の中から図 8 に示す走行シーンの LiDAR データ約 7,500 個を用いた。



図 8 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 9 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。

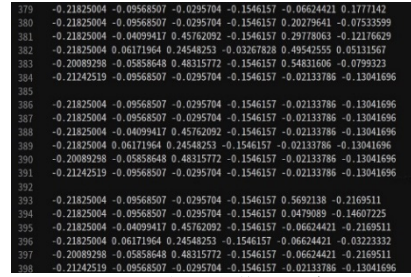


図 9 2次元特徴量(一部)

- (3) 周辺状況検知の結果として得られた特徴量の一部を LSTM に適用した際の精度を図 10 に示す。

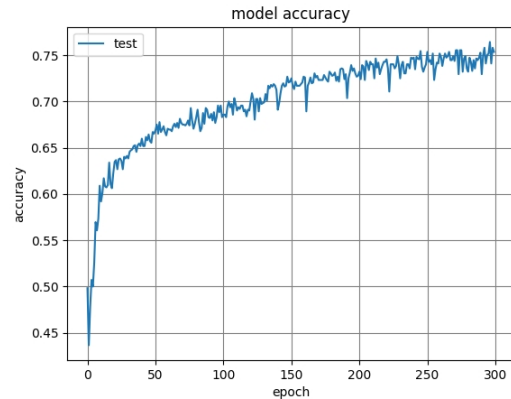


図 10 前方危険予測の精度

- (4) 周辺状況検知の結果として得られた特徴量の一部を LSTM に適用した際の損失関数を図 11 に示す。

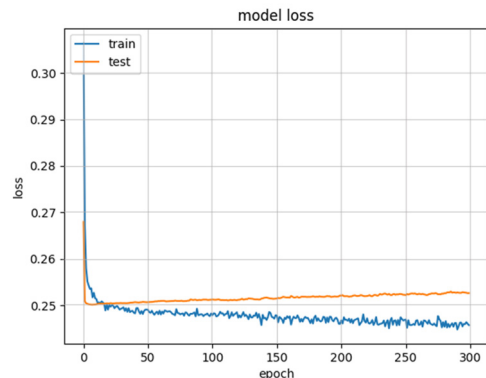


図 11 前方危険予測の損失関数

LSTM を用いた前方危険予測の評価に要した時間を図 12 に示す。

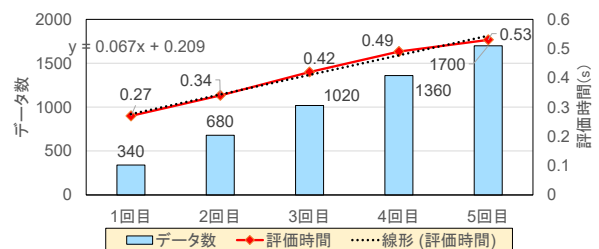


図 12 評価時間

データ数の違いによる前方危険予測の正答率と損失を図 13 に示す。

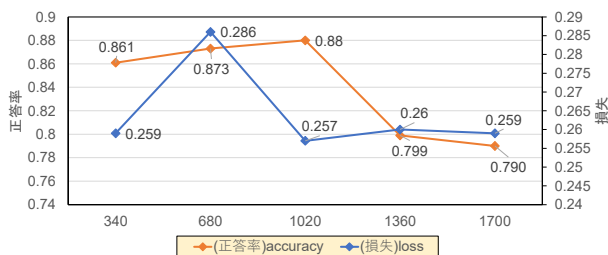


図 13 データ数による正答率と損失

7 評価

7.1 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。

7.2 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 回につき約 10 秒となった。多くの場合で危険予測をできている上、テストでは少ない時間で結果が確認できた。テストデータを用いた前方危険予測に要した時間を図 14 に示す。物体検出については 30 フレームを 1 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 0.2ms で行えるので物体検出後のデータを危険予測に適用した場合でもリアルタイムに予測可能であることが確認できた。



図 14 予測時間

8 考察

8.1 先行研究[5]の結果と本稿との比較

先行研究の結果では、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、プロトタイプの結果では正答率が 70%~80%となっており先行研究と比較して正答率が低い結果となっている。これは学習データ不足による学習不足が原因だと考えられる。理由としては、特徴量のデータ数が 3,119 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。

8.2 評価に基づく提案アーキテクチャ設計方法の考察

先行研究[3]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 CNN で行い、LSTM を用いて危険予測の正答率を測定した。実行結果より 70%~80%の正答率を確認した。この結果から、提案アーキテクチャはカメラを用いた LSTM による事故検知アルゴリズムと同等と位置付けることができる。

8.3 提案方法のスケラビリティ

本稿の提案方法では、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というスケラビリティがあると考えられる。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ができると考える。

9 今後の課題

今後の課題は以下の 3 点である。

(1) 別データ適用時の有用性の評価

本稿では、データセットとして KITTI Dataset を使用して危険予測を行った。そのため、Waymo Open Dataset 等のデータセットを LSTM に適用し、前方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。

(2) 警告システムへの適用

前方危険予測アーキテクチャによって出力された判定結果を警告システムへ入力するアーキテクチャを作成し、適用する必要がある。

(3) シミュレータにおける走行実験

提案した前方危険予測アーキテクチャを CARLA 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。

10 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 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] D.-S. Kim, H.-C. Son, J.-W. Si, and S.-Y. Kim, Traffic Accident Detection Based on Ego Motion and Object Tracking. Journal of AITC, Vol. 10, No. 1, Jul. 2020, pp.15-23.
- [4] A. H. Lang, et al., PointPillars: Fast Encoders for Object Detection from Point Clouds, Proc. of CVPR 2019, IEEE, Jun. 2019, pp. 12697-12705.
- [5] 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.
- [6] 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, May 2016, pp. 441-445.
- [7] 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.