

# AIを用いた交差点での 交通事故危険性要因データ分析

清水 太貴<sup>1</sup>・藤田 素弘<sup>2</sup>

<sup>1</sup>学生会員 名古屋工業大学大学院 工学研究科社会工学専攻学生（〒466-8555 愛知県名古屋市昭和区御器所町）

E-mail: 31415040@stn.nitech.ac.jp

<sup>2</sup>正会員 名古屋工業大学大学院教授 工学研究科（〒466-8555 愛知県名古屋市昭和区御器所町）

E-mail: fujita.motohiro@nitech.ac.jp

我が国における交通事故発生地点の割合は、交差点やその付近で5割を占めており、交差点における効果的・効率的な事故対策を実施していくことが重要となる。現在、交通事故危険性と交差点構造や路面標示との関係性を明らかにするためには、撮影した交差点動画から車両の挙動データを計算している。しかし、この手法は多くの時間を要し大変非効率的である。そこで、AIを用いて自動で車両挙動データを計算することで、様々な交差点からデータを効率的かつ大量に集めることができ、迅速かつ確かな問題把握と対策立案、対策効果の評価が可能になることが期待される。本研究では、AIを用いて自動で動画内を走行する車両の速度、車間距離といった車両挙動データを算出するプログラムを作り、効率的な交通事故危険性要因分析を行うことを目的として行う。

**Key Words :** ai, intersection, vehicle behavior data, deep learning, kalman filter,

## 1. はじめに

近年、AIという言葉がニュースや新聞で耳にする機会が増えてきた。交通の分野においては、車の自動運転システムが有名である。自動運転では物体の現在位置を正確に特定する位置特定技術や、歩行者、走行車両などの動きを検知する認識技術など、AIを用いれば物体から様々な情報を効率的に取得することが可能になると考えられる。交通事故危険性要因分析を行う際に、交差点における車両の走行動画から車両の速度や車間距離を算出する場合は、多くの時間を要するため大変非効率的であり、改善の余地がある。

そこで本研究では、効率的な交通事故危険性要因分析の第一歩としてAI技術を用いて、定点カメラによる動画から走行する車両を認識するとともに、認識した車両から自動で車両挙動データを取得することを試みる。定点カメラによる動画から車両の危険挙動を取れるようにし、車両が危険な動きをする道路や交差点の形状との関係性などを明らかにしていくことを最終目的とする。

## 2. 使用する理論とソフトの詳細

本研究ではディープラーニング<sup>1)</sup>とカルマンフィルタ<sup>2)</sup>のそれぞれ2つの理論を用いて車両の認識を行った。ディープラーニングを用いた車両の認識ではPython3.6<sup>3)</sup>、カルマンフィルタを用いた車両の認識では数学的計算ソフトウェアMATLAB<sup>4)</sup>を使用している。

## 3. 車両挙動データ算出方法

### (1) ディープラーニングを用いた手法

学習済み画像認識モデルVGG16を転移学習させ、車両の認識に特化したモデルを作り、動画内を走行する車両を認識する。認識した車両の周りに四角い枠を発生させ、枠の位置座標などを元に車両挙動データを算出する。本研究では走行速度、車頭時間、車間距離を算出した。

#### a) 走行速度算出方法

動画における任意の位置を、車両を囲う枠が通過し始めた瞬間から通過し終わるまでの動画のフレームの数をカウントし、その値に1フレームあたりに要する時間1/30 [s] を乗算し時間を求める。そしてその時間における移動距離を、車両を囲う枠の進行方向における長さとして、動画内の横断歩道の実長さが4 [m] であることを

用いて求め、これらの値から走行速度を算出する。

b) 車頭時間算出方法

動画における任意の位置を、先に走行する車両を囲うが通過し始めた瞬間から、後続の車両を囲う枠が同じ位置を通過し始める瞬間までの動画のフレームの数をカウントし、1フレームあたりに要する時間1/30 [s] を乗算して、車頭時間を算出する。

c) 車間距離算出方法

動画における任意の位置を、先に走行する車両の枠が通過し終わった瞬間から後続の車両を囲う枠が同じ位置を通過し始めるまでのフレームの数をカウントし、1フレームあたりに要する時間1/30 [s] を乗算し時間を得る。そして後続の車両の走行速度をa)で述べた方法で求め、これらの値を乗算し車間距離を算出する。

(2) カルマンフィルタを用いた手法<sup>2)</sup>

カルマンフィルタは現在の観測値と1ステップ前の状態推定値のみから、現在の状態推定値と1ステップ先の状態予測値を与える反復推定器である。カルマンフィルタを構成するのに必要な数式は大きく分けて3つある。

a) 観測

$$z_k = H_k x_k + v_k \quad (1)$$

$k$  : 時刻

$z_k$  : 観測量

$H_k$  : 状態空間を観測空間に線形写像する観測モデル

$x_k$  : 真値の状態

$v_k$  : 観測誤差

式(1)は観測方程式と呼ばれるものであり、真の状態から推定したいベクトルのみを抽出して、観測誤差を足している。

b) 予測

$$x_k = F_k x_{k-1} + u_k + G_k w_k \quad (2)$$

$F_k$  : システムの時間遷移に関する線形モデル

$u_k$  : 制御入力

$G_k$  : 時間遷移に関する誤差モデルの行列

$w_k$  : 観測誤差

式(2)は現在の時刻の予測推定値を表す。理論的に1つ前の状態から、今の状態を推定している。推定した値はカルマンフィルタによって補正されて、次の入力 $x_{k-1}$ になる。

c) 更新

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + G_k Q_k G_k^t \quad (3)$$

$$S_k = Q_k - H_k P_{k|k-1} H_k^T \quad (4)$$

$$K_k = P_{k|k-1} - H_k^T S_k^{-1} \quad (5)$$

$$e_k = z_k - H_k \hat{x}_{k|k-1} \quad (6)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k e_k \quad (7)$$

$P_{k|k-1}$  : 時刻k-1時点での時刻kの予測誤差

$Q_k$  : 共分散行列

$S_k$  : 観測残差の共分散

$K_k$  : 最適カルマンゲイン

$e_k$  : 観測残差

$\hat{x}_{k|k}$  : 更新された状態の推定値

式(7)がカルマンフィルタによって推定された値になる。1ステップ進むと、この値が $\hat{x}_{k-1}$ となる。観測値と予測値に基づいて車両の位置を推定し、更に観測値と予測値を更新するプロセスを繰り返し行うことで車両の動きを追跡し、追跡した車両から得られるパラメータを用いて車両挙動データ取得を試みる。

4 車両挙動データ取得結果

(1) ディープラーニングを用いた結果

車10台に対してそれぞれ車両挙動データを取得した。算出した走行速度は実際の速度より若干遅くなるという傾向が見られた。

算出した車頭時間は実際の時間とコマ数秒の差がなく、近い結果となった。

算出した車間距離は実際の距離より短くなるという傾向が見られた。

表-1 走行速度

	1	2	3	4	5	6	7	8	9	10
AI[km/h]	43.9	52.1	51.3	47.8	38.9	46.6	50.1	51.2	43.7	48.2
実際[km/h]	46.3	55.4	50.6	46.7	40.0	49.0	50.9	52.4	43.4	49.8

表-2 車頭時間

	1	2	3	4	5	6	7	8	9	10
AI[s]	2.9	1.8	3.4	2.8	6.1	4.2	4.1	3.6	3.5	1.9
実際[s]	2.9	1.7	3.4	2.8	6.1	4.1	4.1	3.6	3.4	1.9

表-3 車間距離

	1	2	3	4	5	6	7	8	9	10
AI[m]	17.1	20.6	22.0	18.1	25.5	26.3	16.8	19.6	22.5	18.5
実際[m]	23.6	25.7	27.5	21.3	32.6	32.4	15.3	23.7	25.2	21.6

走行速度と車間距離で実際の値と差が生じている理由は、車両の動きの量を、認識した車両を囲う枠から得られるパラメータを用いてデータを計算していることだと考えられる。

## (2) カルマンフィルタを用いた結果

現在動画内で動く物体をすべて検出してしまっており、車両のパラメータが得られない状態である。車両のみの動体検出ができるように引き続き画像処理などを行う必要がある。

## 5. まとめ

ディープラーニングを用いて車両から挙動データを取得する方法はカルマンフィルタを用いた方法と比べて、動画内における動体を、車両のみ検出してくることが可能であるが、挙動データを算出する際に用いるパラメータが車両を囲う枠であるので、実際の値と誤差が生じている。

カルマンフィルタを用いて車両から挙動データを取得する方法は、検出してきた車両から直接得られるパラメータを用いて挙動データを算出できることが期待されるが、現在車両以外の動体も検出してしまっている。

今後は動画を二値化した際に、車両以外の動体として検出してしまっている部分を画像処理で除去するなどして改善を行っていく。また、動画内に車両が複数台存在している場合にそれぞれの車両を区別してパラメータが取得できるように検討していく。

また、今回使用した車両の走行する動画は、車両の移動距離を画像のx座標を用いて表すために、車両の真横から撮影したものである。車両がカメラに向かって走行している動画にも対応できるように画像のキャリブレーションについても行っていく必要がある。

今後については、まずは定点カメラによる動画から車両の速度や車間距離などの車両挙動データを算出できるようにする。そして、蛇行運転や一時停止無視などの危険挙動と考えられる走行をする車両を検知できるようにし、車両の危険挙動と道路や交差点構造との関係性を明らかにしていく。

## 参考文献

- 1) 斎藤康毅：ゼロから作るDeep Learning Pythonで学ぶディープラーニングの理論と実装
- 2) 足立修一，丸田一郎：カルマンフィルタの基礎
- 3) 谷岡広樹，康鑫：いちばんやさしいディープラーニング入門教室
- 4) MATLAB ホームページ：https://jp.mathworks.com

(?)

## TRAFFIC ACCIDENT RISK FACTOR DATA ANALYSIS USING AI AT INTERSECTIONS

Taiki SHIMIZU and Motohiro FUJITA

The percentage of traffic accidents in Japan accounts for 50% at and around intersections, and it is important to implement effective and efficient accident countermeasures at intersections. Currently, in order to clarify the relationship between traffic accident risk and intersection structure and road marking, vehicle behavior data is calculated visually from the captured intersection video. However, this method takes a lot of time and is very inefficient. Therefore, by automatically calculating vehicle behavior data using AI, it is possible to collect a lot of data efficiently and in large quantities from various intersections. And It is expected that it will be possible to quickly and accurately identify problems, plan countermeasures, and evaluate the effectiveness of countermeasures. The purpose of this research is to create a program to calculate vehicle behavior data such as the speed and distance between vehicles that automatically move in the video using AI, and to analyze the risk factor of traffic accidents efficiently.