

# Yolo を用いた画像認識による交差点付近における リアルタイム車両状態推定

安齋凌介<sup>†</sup> 伊藤昌毅<sup>††</sup> 大口敬<sup>††</sup> 岩井将行<sup>†</sup>

<sup>†</sup> 東京電機大学大学院未来科学研究科情報メディア学専攻 〒120-8551 東京都足立区千住旭町 5

<sup>††</sup> 東京大学生産技術研究所 〒153-8505 東京都目黒区駒場 4-6-1

E-mail: <sup>†</sup>{anzy,iwai}@cps.im.dendai.ac.jp, <sup>††</sup>{mito,takog}@iis.u-tokyo.ac.jp

あらまし 交通工学はエッジセンサから得られたデータを活用することで、可能性が大きく広がる分野である。速度違反検知、ナンバープレート検知、渋滞検知などは日本のみならず先進国で導入されている。しかしながら、それらのデータを交通の最適化に活用している例はまだ少ない。これは、データの質の低さ、ラベルの欠如、データの利権などが障壁になっていることが考えられる。またそれらのセンサは大規模かつ高額になることが多く、交通量の多い交差点や高速道路に限定して設置されていることが多い。したがって、我々が普段使う道路や交差点ではこれらの導入は遅れていることが現状である。そこで、我々はオープンソースデータと安価なエッジセンサを活用し、交差点付近の車両の状態を推定することができるデバイスを開発した。これにより今まで取ることができなかったその地点の詳細な車両のデータを小規模で安価に取ることが可能となる。

キーワード Yolo, エッジセンサ

## A Real-time Estimation of Vehicle Status Near an Intersection Using Image Recognition by Yolo

Ryosuke ANZAI<sup>†</sup>, Masaki ITO<sup>††</sup>, Kei OGUCHI<sup>††</sup>, and Masayuki IWAI<sup>†</sup>

<sup>†</sup> Department of Information Systems and Multimedia Design, Tokyo Denki University SenjuAsahi-cho 5, Adachi-ku, Tokyo, 120-8851 Japan

<sup>††</sup> Institute of Industrial Science, the University of Tokyo 4-5-6 Komaba, Meguro-ku, 153-8505 Japan

E-mail: <sup>†</sup>{anzy,iwai}@cps.im.dendai.ac.jp, <sup>††</sup>{mito,takog}@iis.u-tokyo.ac.jp

**Abstract** Traffic engineering is a field where the possibilities expand greatly by utilizing data obtained from edge sensors. Speed violation detection, license plate detection, traffic jam detection, etc. have been introduced not only in Japan but also in other developed countries. However, there are still few examples of the use of such data for traffic optimization. This may be due to the low quality of the data, lack of labels, and data rights. In addition, these sensors are often large and expensive, and are often installed only at busy intersections or highways. To solve this problem, we have developed a system using open source data and inexpensive edge sensors. We have developed a device that can estimate the state of vehicles near an intersection using open source data and inexpensive edge sensors. which makes it possible to obtain detailed vehicle data on a small scale at a low cost.

**Key words** Yolo, Edge sensors

### 1. はじめに

IoT ( Internet of Things ) の普及に伴い、さまざまな場所でのセンシングやデータの収集が可能となった。交通工学の分野でも例外ではなく、古くは車両データを扱った技術として速度違反自動取締装置 ( オービス ) が挙げられる。また、近年では ITS

( Intelligent Transport Systems : 高度道路交通システム ) の観点からさらに高度で詳細な交通データを扱う研究やプロジェクトが注目されている。IEEE/CVF Conference on Computer Vision and Pattern Recognition ( CVPR ) Workshops では AI City Challenge [1] と称し、ITS に関するワークショップを毎年開催している。ここでは、カメラによる車両の識別、スピード検知、ナンバープ

レートの読み込みによるマルチカメラでの車両の再識別などのタスクが用意されており、国際的に見ても ITS が注目されていることがわかる。

しかし、現状としてそれらの装置が一般的な道路や交差点に取り付けられていることは少なく、高速道路や交通量の多い幹線道路に限定し取り付けられていることが多い。これは、装置自体が高価で大規模化しやすいことが原因であると考えられる。さらに、日本においてはそれらのデータは、違反や犯罪、渋滞の検知として用いられていることが多く、交通の最適化という観点からデータを活用している事例は少ない。これは、データの質の低さ、ラベルの欠如、データの利権などが障壁になっていると考えられる。したがって、交差点や道路上の車両の状態を推定するには、IoT デバイスの開発を公開されているデータや安価なエッジセンサで行う必要がある。これらのことから本研究では、2 次元のカメラ映像から交差点付近にある複数台の車両の追跡をリアルタイムで行うデバイスの開発を目標とした。

こうしたカメラ画像から複数のオブジェクトを検出し追跡するタスクをコンピュータビジョンや機械学習分野では Multi Object Tracking ( MOT ) と呼ぶ。MOT では、リアルタイムか否かで大きく二つにタスクを分けることができる。リアルタイムではない処理として TrackletNet Tracker ( TNT ) は、2 次元のカメラ映像からオブジェクトを検出し、オブジェクトの軌跡を深層学習を用いて推定していく方法である。<sup>[2][3]</sup> TNT は、カメラ映像内で複雑かつ大量のオブジェクトを追跡することに向いているものの、カメラ映像を全て読み込んで処理するため、リアルタイム性はなく、本研究では用いることができない。一方でリアルタイムでの処理として、Simple Online and Realtime Tracking ( SORT ) がある。<sup>[4][5]</sup> これは、検出されたオブジェクトの座標をカルマンフィルタ<sup>[6]</sup> を用いて、リアルタイムでのオブジェクトの追跡を可能としている。前述で述べた TNT と比べると複雑な MOT を行うことは難しいものの、ある程度規則性を持った車両などの MOT はリアルタイムで処理することが可能である。本研究では、アルゴリズムに SORT、安価なエッジセンサとして Jetson Xavier NX<sup>[7]</sup>、オブジェクト検出機として Yolov4<sup>[8]</sup> を用いて、交差点付近の車両を追跡するシステムを開発した。

本稿では、2 章で提案システムの概要について説明し、3 章ではアルゴリズムについて説明する。4 章では、実際の動作結果を述べ、5 章では、提案システムの有用性や今後の展望について考察する。最後に 6 章でまとめを述べる。

## 2. 提案システム

1 章で述べたように、ITS の観点から安価な IoT デバイスの開発が求められている。このデバイスを開発する際の問題点として、

- (1) 安価であるが故に十分な性能を発揮することが難しいこと
- (2) 既存のデバイスで得られたデータを活用することが難しいこと

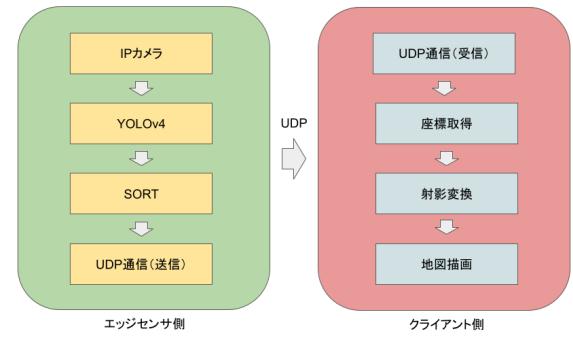


図 1 システム概要

がある。(1) は Jetson Xavier NX を用いて、検証を行う。さらに、SORT は実装が軽量なためリアルタイム処理性能として非常に優れており、本研究ではこれを用いることとした。(2) は The Microsoft Common Objects in COntext (MS COCO) dataset<sup>[9]</sup> を用いて学習された YOLOv4 を使用することで解決する。また、YOLOv4 は画像認識のモデルの中でも検出率及びリアルタイム性に優れており、本研究でこれを用いることとした。

### 2.1 概要

提案する車両の追跡システムの概要は図 1 に示す。提案するシステムはエッジセンサで車両の追跡を行い、クライアント側で座標データを受け取って、地図上に描画する。エッジセンサ側では、検出器に YOLOv4 を用いて、SORT アルゴリズム<sup>[5]</sup> を用いて車両の追跡を行う。クライアント側では、ホモグラフィ行列を求め、Google Maps API<sup>[10]</sup> を用いて地図に車両の位置情報を描画を行う。また、エッジセンサとクライアントの通信は UDP 通信用いている。

### 2.2 システム操作方法

カメラを交差点から約 150m 奥の車両まで映る角度で設置する。クライアント側では、ソフトの起動後エッジセンサから送られてくる画像を元に射影変換するための対応点をカメラ画像と地図でそれぞれ 4 点ずつ設定する。

## 3. アルゴリズム概要

1 章で述べたようにエッジセンサ側で SORT、クライアント側で射影変換を用いている。

### 3.1 SORT について

検出器によって検出したバウンディングボックス ( BBOX ) の座標から、フレーム間の変位を線形モデルを用いて近似させる。各オブジェクトの状態は次のようにモデル化される。

$$X = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T \quad (1)$$

ここで、 $u$  と  $v$  はオブジェクトの中心の水平方向と垂直方向のピクセル位置を表し、 $s$  と  $r$  はそれぞれオブジェクトの BBOX の面積とアスペクト比を表す。ただしアスペクト比は一定とする。検出された BBOX は、カルマンフィルタのフレームワークを用いて、速度成分が最適化されるオブジェクトの状態を更新するために用いられる。オブジェクトが検出されていない場合、線形モデルを用いて補正せずに、次の状態を単純に予測する。

### Algorithm 1 SORT

```

1: 動画の読み込み
2: while do
3:   フレームの読み込み
4:   YOLOv4 でオブジェクト探索
5:   if Object is True then
6:      $X = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T$ 
7:     kalmanfilter( $X$ )
8:   end if
9: end while

```

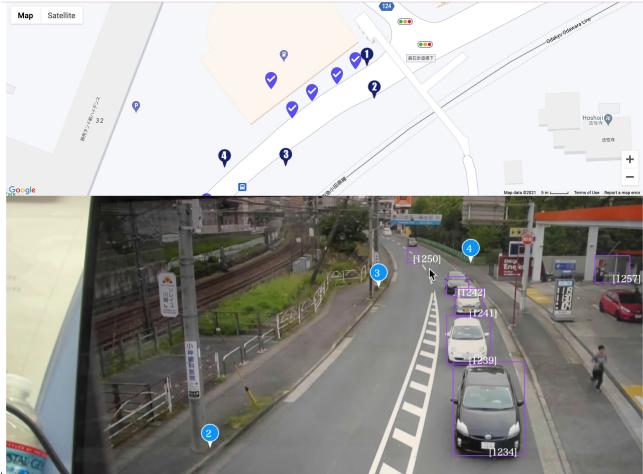


図 2 クライアント側のデモシーン

### 3.2 射影変換

検出器によって検出した BBOX の底辺の中点  $p$  を射影変換し、地図上に投影する。中点  $p$  は次式のようになる。

$$p = [x, y, 1]^T, P = [X, Y, 1]^T \quad (2)$$

$$P' = [X', Y', W']^T = H_p \quad (3)$$

$$P = \frac{1}{W'} P' \quad (4)$$

ここで、 $P$  は地図上の座標を表している。 $x, y$  は画像の座標を表しており、 $H$  はホモグラフィ行列を表す。

## 4. 検証結果

安価であるが十分な性能で追跡を行えているかを評価とする。

### 4.1 検証環境

検証環境として、図 3 のように web カメラを交差点付近の映像が映し出されている画面の前に置き、擬似的にリアルタイム環境にした。また、カメラの解像度は 1080p とした。

### 4.2 結果

図 2 を見たい。これはクライアント側で描画したものである。地図上部に表示されているチェックマークが各車両の位置を示している。リアルタイムで車両の位置を追跡し、動的に表示が変わっている。また、地図下部に表示されているのは実際にカメラから送られてきた画像が BBOX が描画され



図 3 配置構成：左画面はクライアントの画面

```

FPS = 2.086570
2021-02-24T07:47:48.712231,526,14
2021-02-24T07:47:48.712231,1715,1
2021-02-24T07:47:48.712231,1717,1
2021-02-24T07:47:48.712231,1740,1
2021-02-24T07:47:48.712231,1746,1
2021-02-24T07:47:48.712231,1765,9
2021-02-24T07:47:48.712231,1774,9
FPS = 2.083085
save data
2021-02-24T07:47:49.195478,526,14
2021-02-24T07:47:49.195478,1715,1
2021-02-24T07:47:49.195478,1717,1
2021-02-24T07:47:49.195478,1740,1
2021-02-24T07:47:49.195478,1746,1
2021-02-24T07:47:49.195478,1765,9
2021-02-24T07:47:49.195478,1774,9
FPS = 1.888533
2021-02-24T07:47:49.788101,526,14
2021-02-24T07:47:49.788101,1715,1
2021-02-24T07:47:49.788101,1717,9
2021-02-24T07:47:49.788101,1746,1
2021-02-24T07:47:49.788101,1749,1
2021-02-24T07:47:49.788101,1765,9
2021-02-24T07:47:49.788101,1774,9

```

図 4 エッジセンサ側のターミナル画面

ている。

図 4 より FPS は 2.0 ~ 3.0 で推移した。ただし、クライアント側に送るカメラ画像の生成時は 1.5 ~ 2.0 になった。カメラの解像度を下げることで、FPS の値が上がることが確認できるが、オブジェクトの検出率が下がることを確認した。また、映像内におけるオブジェクト数が増えると FPS が低下することを確認した。映像を直接読み込むよりも IP カメラを用いた方がオブジェクトの認識が悪いことが確認できた。

## 5. 考 察

結果より、車両の追跡を行える程度の性能であることを確認した。本研究は特に信号機の制御の最適化を行うためのセンサとして開発しているため、交差点から約150mまでの車両の状態を検知する必要がある。しかし、検証では、カメラから離れば離れるほどオブジェクトの認識率は低くなり、その精度も下がることがわかった。これは、検証環境の都合上、画面に投影された映像をカメラで撮り、SORTの処理を行っていたためで、結果からも、映像を直接読み込んだ場合の方が認識率が上がる事がわかった。また、FPSとオブジェクトの検出数には反比例の関係にあるため、今後カメラの画素数などのパラメータを調整する必要がある。

## 6. ま と め

本研究において安価なエッジセンサとしてJetson Xavier NXを用いて、その性能について検証を行った。1章で述べたAI City Challenge 2021に参加する前段階として、本研究の進捗を本稿にまとめた。現状では車両の追跡までに止まっており、今後はワインカーから右左折の判定や、スピードの検知を行えるエッジセンサの開発を目指す。また地図上に表示されている車両のIDが表示されておらず、どの車両であるのかがわかりづらくなっているため、今後改良する必要がある。

## 謝 辞

本研究は、一般財団法人トヨタ・モビリティ基金の支援による「自律分散型信号システム研究開発」の一部として実施した。

## 文 献

- [1] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Xiaodong Yang, Liang Zheng, Anuj Sharma, Rama Chellappa, and Pranamesh Chakraborty. The 4th ai city challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.
- [2] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. In *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 482–490, 2019.
- [3] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 108–115, 2018.
- [4] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pp. 3464–3468. IEEE, 2016.
- [5] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pp. 3645–3649. IEEE, 2017.
- [6] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.
- [7] NVIDIA. Jetson xavier nx, 2021. <https://www.nvidia.com/ja-jp/autonomous-machines/embedded-systems/jetson-xavier-nx/>.
- [8] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
- [9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.
- [10] Google. Google maps platform. <https://developers.google.com/maps/?hl=ja>.