

画像認識を用いた交通量調査

2020SC021 今西健

指導教員：大石泰章

1 はじめに

交通量調査は全国の道路交通の現状と問題点を把握し、将来に渡る道路の整備計画を策定する資料を得るために行われている [1]. 交通情報を自動で収集し処理することの利点は、散発的に行われている交通量調査をリアルタイムに行えるということ、そして、一定の基準にもとづく正確で品質の安定した調査結果を得ることができるということである。先行研究では、画像処理を用いて交通流に関するデータ収集を行う研究がなされている [2].

本研究は、画像処理を用いて、自動車の台数を自動で数え、道路の混雑度を測定することを目的とする。具体的には、深層学習により短時間で物体の検出を行うことのできる You Only Look Once (YOLO) というソフトウェアを用いて車の検出を行い、道路の混雑度を自動で測定する。

2 使用する環境

2.1 使用する機器

本研究では、PC と、PC 上に実装された You Only Look Once version 5 (以下 YOLOv5 と記述する) と Python 3.11.5, およびカメラを使用する。

YOLOv5 は処理速度が非常に速い物体検出アルゴリズムを実装したソフトウェアである。Open CV は、インテル社のエンジニアが開発した、無料で配布されている画像処理、画像解析のためのライブラリである。YOLOv5, Open CV は Python 上で運用することが可能であり、ライセンスフリーで使用できる。カメラは、eMeet Nova USB WEBCAM と iPhone 15 Pro Max を使用する。eMeet はオートフォーカスで 1920 × 1080 画素のセンサを搭載している。カメラを図 1 のように設置し、車の撮影を行う。カメラで撮影した画像は PC に転送され、次に述べる YOLOv5, および Open CV で処理される。



図 1 実験の様子

2.2 YOLOv5

YOLO は Redmon, Divvala, 他によって開発された物体検出アルゴリズムである [3]. YOLO は被写体の範囲を特定するためにバウンディング・ボックスという長方形の図形を用いる。このバウンディング・ボックスとグリッドセルを用いることにより YOLO は極めて正確性の高い推測技術を持つ。しかし、グリッド内で識別可能なクラスは 1 つであり、検出可能な物体の数は 2 と制約があることから、グリッド内に大量のオブジェクトがある画像の場合は物体検出の精度が低くなる。

2.3 OpenCV

OpenCV を導入すると、画像の認識、画像の編集、物体の検出、テンプレートマッチングが利用可能となる。本研究では、画像の認識、画像の編集の機能を利用する。

3 構築するシステム

3.1 YOLO の仕様

YOLO を使用しデータを解析すると、ニューラルネットワークを利用し物体の領域を識別、出力する。動画データを YOLO に読み込むと、動画の画素サイズは 640 × 640 のサイズに縮小され、フレームレートは毎秒 30 フレームとなる [4].

3.2 YOLO の距離限界

YOLO には複数の物体が重なっている状況では物体を判別できないという特徴がある。本研究では、地面から高さのある歩道橋で対象を撮影する。真上からの撮影ではないため、カメラからの距離が離れてしまうと対象が重なってしまい、YOLO は正しく機能しなく借る距離を調べた。

複数の条件下で YOLO に動画および静止画を読み込ませ、車の識別を行った。歩道橋の上で 13 時頃と 21 時頃に撮影を行った。昼間と夜間の事例をそれぞれ図 2 および、図 3 に示す。

カメラが車の正面を撮影できる状態で、動画および静止画の撮影と物体の認識を行った結果、車とカメラの距離が約 40m の範囲内であれば誤差が少なく車の検出が可能であることが分かった。カメラとの距離が 40m を超えると本システムでは対象が小さく、かつ対象が重なった状態で撮影されてしまうため車の検出は困難となる。これは、YOLO が被写体の範囲を特定するために用いるバウンディング・ボックスの特性が関係している。40m 以上の範囲はグリッド内に多数の目標が存在している状況であるため物体検出の精度が著しく低下する。

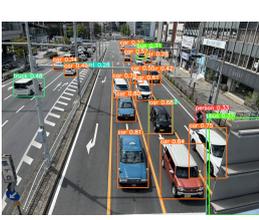


図2 昼間の撮影

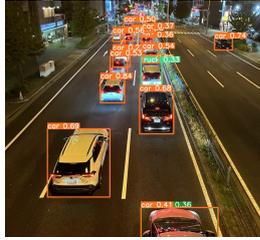


図3 夜間の撮影

3.3 構築するシステム

YOLO の仕様を考慮して、システムを構築する。

最初に、カメラで撮影した画像データを PC に送信する。そして、Open CV を用いて画像の編集を行う。YOLO は 40m を超えると検出精度が著しく低下する。誤検出を防ぐために YOLO の検出精度が低下する 40m を超えた地点の画像をトリミングすることで正確なデータを算出できるようにする。そして、トリミングした画像に映った車を YOLOv5 を用いて、それぞれバス、トラック、乗用車の三種類に分けて検出する [4]。その検出結果から、画像中のそれぞれの車の台数を特定する。そして、その結果を Open CV を用いて画像上とコマンドプロンプト上に表示する。画像上、およびコマンドプロンプト上に表示する情報はその瞬間における乗用車、トラック、バスのそれぞれの台数および、それを合計した台数を表示する。

4 実験

4.1 実験方法

作成したプログラムを用いて実験を行った。それぞれ 13 時頃、21 時頃の二つの時間帯で動画を撮影した。動画は画素数が 1080 画素、毎秒 60 フレームで約 20 秒の動画である。1 秒ごとに乗用車、トラック、バスの台数を求め、その変化の様子を図 4 および図 5 に示す。

4.2 実験結果

時間経過に応じて車の台数が変化していることが図 4、図 5 から読みとれる。動画を目視した結果と比較し、正しくプログラムが動作していることを確認した。車の種別も正しく判別し、検出することができる。

夜間において、アルファードやベルファイヤなどの大型の乗用車をトラックと誤認識してしまう場合が発生した。カメラの画角の外から車が侵入してきた際に一瞬、YOLO は大型の乗用車をトラックと誤認識していた。これは、ライトなどの逆光がカメラに影響して、YOLO の学習済みのモデルと対象が異なって撮影されてしまうことから発生すると考えられる。

5 おわりに

本研究では、YOLOv5 および Open CV を用いて車を認識し、台数のカウントを行った。その結果、その瞬間に

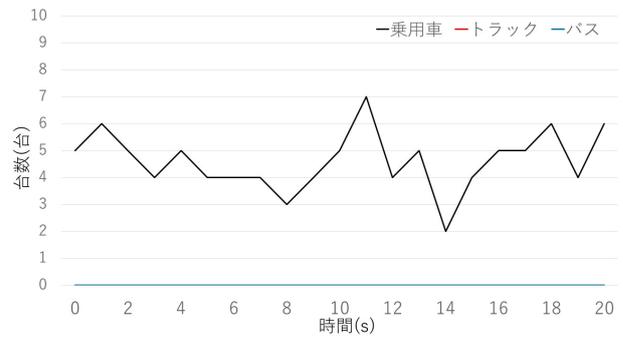


図4 昼間の車の台数の変化

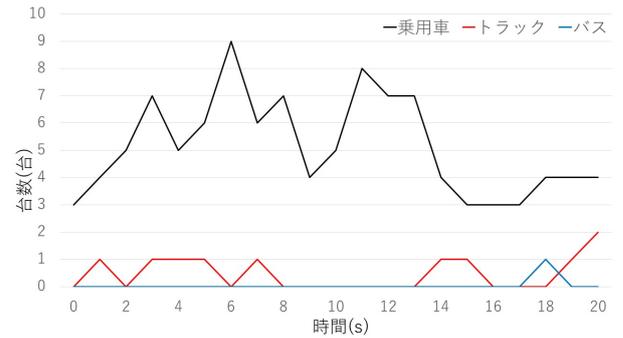


図5 夜間の車の台数の変化

ある地点を通過する車の台数を正確に判別することが出来た。一方、一定時間の交通量を算出することは出来なかった。これを行うためには対象が撮影されている間、その対象が同一であるかを判別する技術を導入しなくてはならない為である。これを行うことは、今後の課題である。また、自動車の専有面積や向きから、より正確な混雑度を求める必要があると考える。自動車の向きを判別することができれば、一台のカメラでそれぞれの車線の交通量を測定する事ができる。

参考文献

- [1] 国土交通省:一般交通量調査について. <https://www.mlit.go.jp/road/census/h22-1/data/kasyorep.pdf>
- [2] 鈴木 一史, 中村 英樹: 交通流解析のためのビデオ画像処理システム TrafficAnalyzer の開発と性能検証, 土木学会論文集, Vol. 62, No. 3, pp. 276-287, 2006.
- [3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi: *You only look once: unified, real-time object detection*, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, June 2016, pp. ~779-788.
- [4] G.Jocher:YOLOv5 by ultralytics,2020. <https://github.com/ultralytics/yolov5>