

DeepCounter：深層学習を用いた細粒度な ゴミ排出量データ収集手法

三上 量弘^{1,a)} 陳 寅^{1,b)} 中澤 仁^{1,c)}

受付日 2019年4月5日, 採録日 2019年10月3日

概要：ゴミの排出量は、住民の家族構成やその生活パターンに影響され、都市そのものの変化を反映する。したがって、ゴミ排出量データが取得可能となれば、清掃車のルート最適化や排出量削減のための啓蒙など、様々な行政施策に活用可能と考えられる。しかし、現在取得可能なゴミ排出量データは清掃車ごとの回収量にとどまっており、粒度が粗く細微な都市の変化をとらえることは困難である。ゴミは排出量に基づく多様な行政施策を可能とするには、回収地点や量を含む細粒度データが必要となるが、そのようなデータを取得する実際的な手法は欠如している。そこで、本論文では DeepCounter を提案する。DeepCounter は清掃車に搭載され、清掃車の後方動画に深層学習による物体検出技術を適用することで、清掃車が回収作業をしながらリアルタイムに回収したゴミ袋の個数を推定するセンシングシステムである。実験では藤沢市を走行する清掃車の後部ドライブレコーダで撮影した動画を用いて、提案手法の有効性を検証した。

キーワード：深層学習, 画像処理, スマートシティ, automotive sensing

DeepCounter: Collecting Garbage Disposal Data with Deep Learning

KAZUHIRO MIKAMI^{1,a)} YIN CHEN^{1,b)} JIN NAKAZAWA^{1,c)}

Received: April 5, 2019, Accepted: October 3, 2019

Abstract: Various kinds of waste are generated in our daily life, making garbage collection one of the most fundamental public services in modern cities. The amount of waste disposal varies significantly in both spatial and temporal dimensions, most probably resulting from the structure of inhabitants and their lifestyle. In other words, changes in the spatio-temporal distribution of garbage disposal also reflect the changes in the city itself. Such fine-grained information on the spatio-temporal distributions of garbage disposal is expected to be utilized for novel applications in waste collection service, urban planning and garbage reduction. However, the lack of an effective yet financially practical approach to automatically recognizing the category and measuring the amount of the collected garbage remains a major obstacle to fulfill the appealing vision of the applications. In this paper, we propose DeepCounter, an automotive sensing system mounted on garbage trucks where the videos of garbage collection are processed for detection of the category such like burnable or non-burnable and amount of collected garbage bags of the corresponding category. A detection-tracking-counting (DTC) algorithm for collecting the data is developed. Experimental evaluation validates the feasibility of the proposed approach using video of realistic garbage collection in Fujisawa city, Japan.

Keywords: deep learning, image processing, smart city, automotive sensing,

1. はじめに

人が生活を営むうえでゴミは発生する。そのため、ゴミ回収業務は都市において重要な公共サービスの1つとなっている。図1は、神奈川県藤沢市のゴミ排出量に関する

¹ 慶應義塾大学
Keio University, Fujisawa, Kanagawa 252-0882, Japan
a) quantan@ht.sfc.keio.ac.jp
b) yin@ht.sfc.keio.ac.jp
c) jin@ht.sfc.keio.ac.jp

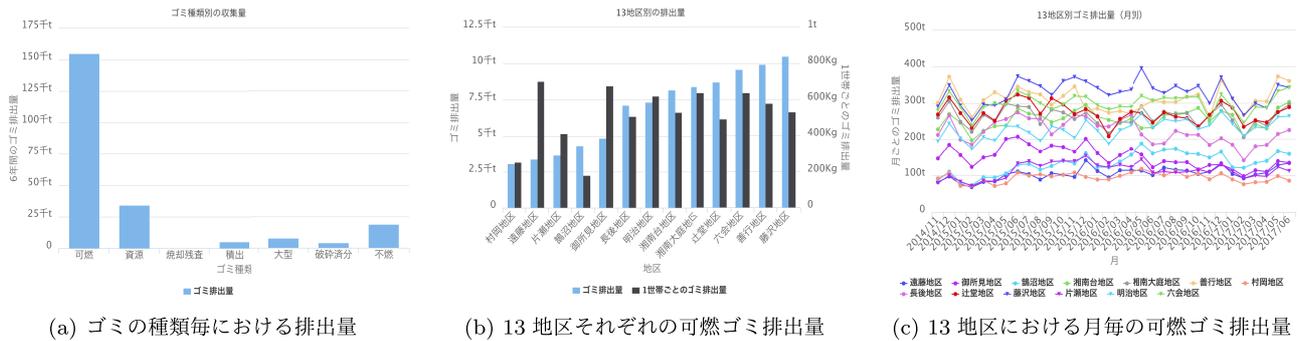


図 1 藤沢市の様々なゴミの排出量データ [1]

Fig. 1 The spatio-temporal distribution of waste disposal of Fujisawa city from Nov. 2014 to Jun. 2017 [1].

データである。図 1(b) から、郊外の遠藤地区では 1 世帯あたりのゴミ排出量が多い一方、都市中心部の藤沢地区では少ない。また、図 1(c) から、ゴミ排出量は 2 月ごろに減少し、6 月ごろに増加している。これらのデータは、住民の家族構成やその生活パターンがゴミ排出量に影響を与えることを示す。つまり、ゴミ排出量は、都市そのものの変化を反映する。したがって、ゴミ排出量データが取得可能となれば、清掃車のルート最適化や排出量削減のための啓蒙など、様々な行政施策に活用可能と考えられる。

しかし、現在取得可能なゴミ排出量データは清掃車ごとの回収量にとどまっており、粒度が粗く微細な都市の変化をとらえることは困難である。ゴミ排出量に基づく多様な行政施策を可能とするには、回収地点や量を含む細粒度データが必要となるが、そのようなデータを取得する実際的な手法は欠如している。たとえば、ゴミの排出量は回収作業員や住民によって測定することも可能ではあるが、費やす労力、時間、金銭のコストは自治体や住人にとって大きな負担となるため、細粒度なゴミ排出量を推定可能なデータ収集手法が必要である。

そこで、本研究では、回収したゴミ袋の個数を自動的に推定するシステム、DeepCounter を提案する。DeepCounter は清掃車に搭載され、清掃車の後方動画に深層学習による物体検出技術を適用することで、清掃車が回収作業をしながらリアルタイムに回収したゴミ袋の個数を推定するセンシングシステムである。ゴミ袋は Detection-Tracking-Counting アルゴリズム (DTC) によって計数される。DTC は動画に写るゴミ袋を自動的に検出し、追跡、計数できる。通常、清掃車には事故を記録するため後部にドライブレコーダが取り付けられている。しかし、ドライブレコーダによって撮影された動画は他の目的に活用されていない。その動画を活用することで、ゴミ回収とゴミ排出量データの収集を同時に行える。評価実験では藤沢市を走行する清掃車の後部ドライブレコーダで撮影した動画 (以後、ゴミ回収動画と呼ぶ) を用いて、提案手法の有効性を検証した。

本研究の先行研究 [2] では、DTC の検出プロセスにベ-

スネットワークが 13 層からなる Single Shot Multibox Detector (SSD) を採用し、検出精度を表す Average Precision (AP) において 61.9, および処理速度を表す FPS において 6.2 を達成した。本論文では、新たなデータセット作成と検出プロセスの改善を行い、評価実験を再実施し、AP において 78.9, および FPS において 15.2 を達成した。また、評価実験から得られた新たな知見をまとめた。

本論文の構成は以下のとおりである。2 章では、自動車にセンサを搭載してセンシングを行う Automotive Sensing と、深層学習を用いた物体検出手法の関連研究について述べる。3 章では、本研究で提案するゴミ排出量センシングシステム、DeepCounter を概説する。4 章では、ゴミ袋の個数を計数するアルゴリズム、DTC を詳述する。5 章では、DTC のゴミ袋検出精度、処理速度と計数精度の実験と結果を示したうえで、実験結果についての考察を行い、6 章で本論文の結論を述べる。

2. 関連研究

2.1 Automotive Sensing

Automotive Sensing (AS) は自動車に GPS やドライブレコーダ、加速度計など様々なセンサを搭載することで、都市のセンシングを行いデータを収集する。無線通信やセンサデバイス、深層学習の発展により、これらの技術を組み合わせることで、多様なデータの取得が可能となった。これまでに、AS は交通状態のモニタリングなど、多くの場所に適用されている。Chen らは、AS の時空間的カバレッジの定義を新たに提案し、清掃車に設置した GPS で取得したデータを用いて評価した [3]。Rezaei らは GPS、車輪の速度、ステアリングの角度、ジャイロスコープなどのセンサデバイスによって取得したデータを入力として、GPS が使用不可能な状況でも、位置の推定が可能なアルゴリズムを提案した [4]。Fujii らは GPS による位置情報と車間距離を測定可能なセンサデバイスを用いてデータを収集し、そのデータを近隣の車どうしで共有することで、位置の推定が可能なシステムを提案し、GPS のみの位置推定よりも

85% 精度の高い推定を行えることを示した [5].

また AS は、都市の状態を検知するためにも広く利用されてきた. Eriksson らは、タクシーに搭載した加速度センサで取得したデータを機械学習のモデルに学習させ、道路表面のくぼみを検知するシステムを提案した [6]. Kawano らは、清掃車に搭載したドライブレコーダで撮影した動画を畳み込みニューラルネットワーク (CNN) に学習させることで、掠れた路面標識を検出するシステムを提案した [7]. Du らは、自動車を動的なセンサとして交通情報を取得し、都市の交通状況をモニタリングするシステムを提案した [8].

AS は、細粒度な環境情報をモニタリングするためにも利用される. Hasenfratz らは、走行する公共交通機関の車両に大気汚染を測定可能なセンサを搭載し、取得したデータから大気汚染マップを構築する回帰モデルを提案した [9]. Chen らは、ゴミを戸別回収する藤沢市の清掃車に PM2.5 センサを搭載し、3G 通信システムを用いることで、リアルタイムな PM2.5 のデータを住民へ提供するシステムを提案した [10]. Kishino らは清掃車に搭載したモーションセンサを用いてゴミの量を推定する手法を提案した [11]. しかし、この手法では清掃車が回収する領域全体におけるゴミの排出量を推定することは可能だが、各ゴミ回収地点のような細粒度なゴミ排出量データを取得することはできない. 収集可能なデータに限りはあるものの、様々なセンサデバイスと機械学習、無線通信技術を組み合わせることにより、多様なデータを収集し、利活用する可能性を AS は示している.

2.2 深層学習による物体検出

深層学習による物体検出が登場する以前は、人が設計した特徴量を利用した物体検出が主流だった. Viola らは画素の明暗を利用した Haar 特徴量を提案し、少ない計算量で高い顔検出精度を達成した [12]. Dalal らは画素の勾配を利用した HOG 特徴量を提案し、人物検出において既存手法を大きく上回る精度を達成した [13].

しかしながら、すべての物体の特徴をとらえ検出できる網羅的特徴量を人が設計するのは困難である. 一方、深層学習は自動的に特徴量を獲得し、層を深めることで特徴量の多様性を表現できる. そのため、人が設計した特徴量と比較し、高精度かつ多様な物体検出ができる.

深層学習による物体検出は CNN を使用し、主に 2 段階と 1 段階で物体を検出する手法の 2 つに分けられる. 一般的に、2 段階手法では物体検出の精度で優れており、1 段階手法は検出の処理速度において優れている. 表 1 に主な物体検出フレームワークの精度と処理速度を示す. mAP (mean Average Precision) と FPS (Frame per second) はそれぞれ物体検出の精度と処理速度を表す. mAP は物体検出のコンペティションである PASCAL VOC [14] の 2007

表 1 各物体検出フレームワークの精度と処理速度の比較 [15]

Table 1 Comparison of accuracy and processing speed of each object detection framework [15].

物体検出のフレームワーク	手法	Train	mAP	FPS
R-CNN (VGG16)	2 段階	2012	66.0	0.03
Fast R-CNN	2 段階	2007+2012	70.0	0.5
Faster R-CNN (VGG16)	2 段階	2007+2012	73.2	9.1
Faster R-CNN (ResNet101)	2 段階	2007+2012	83.8	0.4
YOLO v1	1 段階	2007+2012	57.9	45
SSD (300)	1 段階	2007+2012	74.3	46
SSD (500)	1 段階	2007+2012	76.8	19
YOLO v2 (554 x 554)	1 段階	2007+2012	78.2	40

年のテストデータセットを用いて評価されている.

2.2.1 2 段階手法

2 段階手法では、第 1 段階で物体と推定した領域の候補を生成し、次の段階で CNN を用いて推定した候補から正しい領域選択とクラス分類を行う. Girshick らが提案した R-CNN では、まず入力画像にある物体を selective search [16] による物体の領域推定を行い、次いで CNN によって物体の特徴量抽出し、その特徴量を使って複数の SVM により正確な物体領域の検出と分類を比較的精度良く行う [17]. 一方で、R-CNN は物体検出を多数の機構で行っていたことに加え、すべての物体領域の候補に対して CNN を適用していたため、計算コストが高かった.

この問題を改善したのが、Fast R-CNN [18] である. Fast R-CNN では、画像そのものを CNN へ渡して特徴量抽出を行い、その特徴量から物体領域の候補を提案するなどにより、R-CNN と比較して物体検出精度と処理速度両方において改善された. Fast R-CNN をさらに改良した Faster R-CNN [19] では、Fast R-CNN で処理速度のボトルネックとなっていた selective search による物体領域候補の生成を、CNN を用いた Region Proposal Network へと変更することにより Fast R-CNN より高い mAP を達成しながらも 18 倍の処理速度を達成している. しかし、いずれの 2 段階手法においても、まず物体領域の候補を生成し、その候補の中から正しい候補を選択することによって物体検出を行っているため、検出精度は高いものの、検出する処理に多くの時間を必要とした. この問題を解決するために考案されたのが 1 段階手法である.

2.2.2 1 段階手法

1 段階手法では、2 段階手法では分けられていた物体領域候補の生成と正しい候補の選択を、1 つの CNN で同時に行う. これにより、2 段階手法より高速な処理を実現している. Redmon らによって提案された YOLO [20] は、単一の CNN によって物体領域の推定と物体のクラス分類を行うことによって物体検出を実現している. 表 1 が示すように、YOLO は Faster R-CNN と比較すると検出精度は劣りながらも大幅な処理速度の高速化を達成している.

SSD [21] は、単一の CNN で特徴量抽出を行い、大きさの異なる複数の特徴マップと Multibox [22] による物体領域推定を組み合わせることにより、画像に写る大きさの異なる様々な物体に対して自然な検出を実現している。これにより SSD(300) は、YOLO よりも高速な検出速度を実現しながらも、Faster R-CNN よりも高い検出精度を実現している。YOLOv2, YOLOv3 [23], [24] や Retinanet [25], DSSD [26] のように、より高速な検出速度かつ高精度な新しい物体検出システムが継続的に提案されている。また、RefineDet [27] のように 1 段階手法でありながらも 2 段階手法の良いところを取り入れた手法も提案されている。1 段階手法により、高精度かつ高速度での物体検出が可能となった。

3. DeepCounter : ゴミ排出量データセンシングシステム

3.1 藤沢市におけるゴミ回収

ゴミ回収は、定められた集積場に近隣の住宅がゴミを出し一括で回収する方法や、各戸や集合住宅の前にゴミを出してもらい、戸別ごとに回収する方法などがある。藤沢市では、家庭のゴミは戸別かつ日ごとに回収している。図 2 のように、各家庭や集合住宅は、ゴミ袋を入れるためのバケツやコンテナを設置している。図 2 (b), 図 2 (c), 図 2 (d) のように、住民はそれぞれ定められた色のゴミ袋を使用し、可燃ゴミ、不可燃ゴミ、プラスチックなどゴミを分別する。そして藤沢市により定められたスケジュールに従って、種類に応じた回収日に清掃車が回収していく。ゴミの回収地点に清掃車が到着すると、ゴミ回収作業員はトラックから降りてコンテナに移動し、ゴミ袋を清掃車のトランクに投げ入れていく。投げ込まれるプロセスは、清掃車後部に設置したドライブレコーダで撮影され、その動画に深層学習による物体検出技術を用いることで、回収したゴミを自動的に検知できる。ゴミ袋は戸別かつ種類ごとに回収するので、GPS によって取得した位置と時刻を付与することで、細粒度な種類ごとのゴミ排出量の時空間データを取得することが可能となる。ゴミ排出量データ取得プロセスを図 3 に示す。

3.2 DeepCounter のシステム構成

図 4 に示すように、DeepCounter はドライブレコーダ、GPS レシーバ、組み込みコンピュータ、および無線通信デバイスで構成される。DTC は組み込みコンピュータ上でゴミ回収動画を処理し、ゴミ袋の個数を計数する。計数したゴミの個数は、時刻と位置情報とともに、通信ネットワークを介してデータサーバへ送信される。なお、プライバシー保護の観点から、撮影された動画データはデータサーバに送信しない。本研究では、組み込み型コンピュータとして、NVIDIA Jetson TX2 開発キットを採用した。Jetson TX2



図 2 藤沢市におけるゴミの回収

Fig. 2 An example of house waste classification of Fujisawa city, Japan.



図 3 ゴミ排出量データ収集プロセス

Fig. 3 The garbage collection in Japan.

には GPU が搭載されているため、深層学習による物体検出を高速に行える。そのため清掃車でのエッジ処理が可能となり、清掃車が回収作業をしながらリアルタイムでゴミの排出量データを取得できる。

リアルタイムでデータを取得する利点は 2 つある。1 つはコスト削減である。リアルタイムに清掃車でエッジ処理を行うことで、動画データを保存する設備コストに加え、送信する通信コストなどが不要になる。2 つ目はプライバシーの高さである。動画データの保存・通信が不必要になるため、付随する住民のプライバシー情報の流出が妨げられる。

4. Detection-Tracking-Counting アルゴリズム (DTC)

本章では、DeepCounter 上でゴミ袋を計数する DTC について述べる。図 5 に示すように、DTC は Detection, Tracking, Counting の 3 つのプロセスで構成される。それ

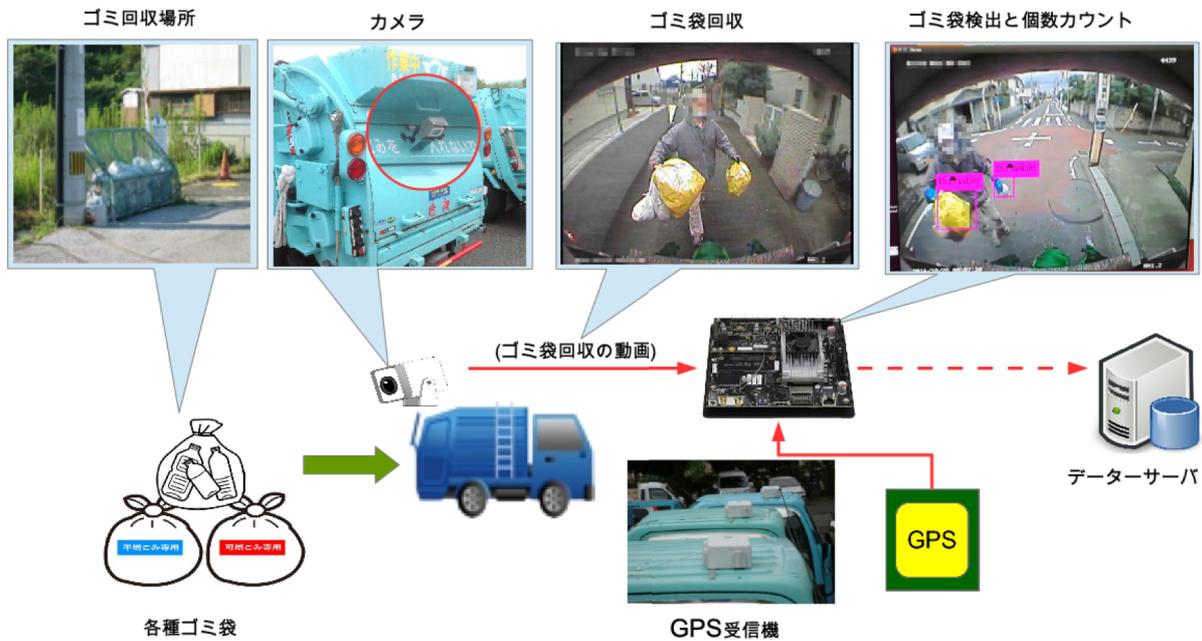


図 4 DeepCounter のアーキテクチャ
 Fig. 4 The architecture of DeepCounter.

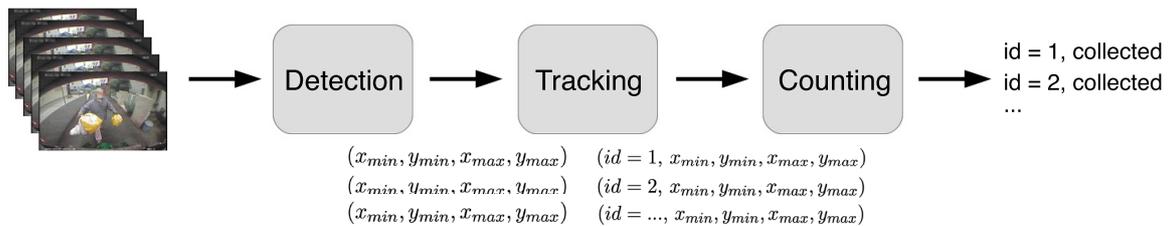


図 5 DTC のプロセス
 Fig. 5 The DTC algorithm.

それぞれの機能は以下のように定義される。

- (1) Detection: ゴミ回収動画の各フレーム内に写るゴミ袋の位置を検出。
- (2) Tracking: 動画の各フレーム間でゴミ袋を同定。
- (3) Counting: 定められた条件によりゴミ袋の回収判定を行い、計数する。

4.1 Detection プロセス

Detection プロセスでは、動画の各フレームに写るゴミ袋の位置を検出する。Detection プロセスにおけるゴミ袋の検出は CNN をベースとした深層学習による物体検出で行う。DTC では Liu らにより提案されている Single shot Multi Box Detector (SSD) [21] をカスタマイズしたものをを用いる。表 1 が示すように、SSD は検出精度、処理速度ともに優れている。SSD は、VGG-16 [28] を特徴量抽出のためのベースネットワークとして使用している (図 6)。しかし、VGG-16 は計算コストが高く、高速な処理が求められる DeepCounter には適さない。

そこで、DTC では、VGG-16 を参考に、ベースネットワークの層を削減することで処理の高速化を図った。その

一例として、5 層のベースネットワークからなる SSD のアーキテクチャを図 7 に示す。DTC における SSD は、ゴミ回収動画のフレームを入力とし、検出したゴミ袋の画像における位置を表す Bounding Box (BB) と、検出した物体がゴミ袋である確率を出力する。Detection のプロセスでは、検出した物体を閾値 τ_c 以上の確率でゴミ袋であると判定した場合、その BB データを Tracking プロセスへ送る。表 2 に DTC のパラメータを示す。

4.2 Tracking プロセス

動画に映る物体を Tracking するアルゴリズムとして、カラーヒストグラムを用いる手法がある [29]。しかし、Detection プロセスが出力するのはゴミ袋の位置情報であるため、カラーヒストグラムを有効活用することは難しい。そこで本研究では、Tracking プロセスにおいてフレーム間におけるゴミ袋の同一性を判定する手法として Intersection over Union (IoU) に基づく判定基準を採用する。図 9 に示すように、IoU は 2 つの領域の和集合と積集合の比率として定義される。

Tracking プロセスは Detection プロセスより検出したゴ

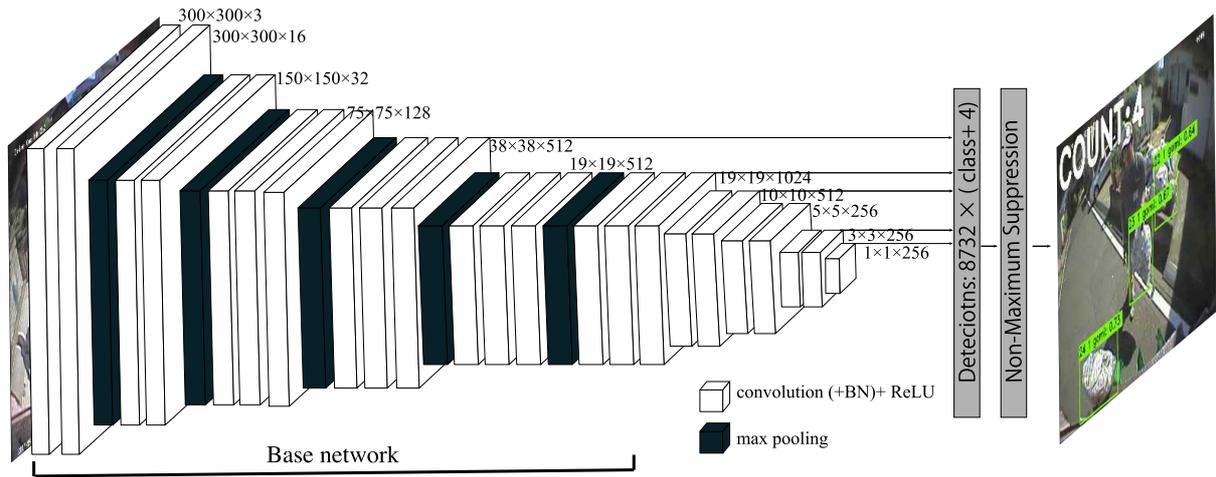


図 6 ベースネットワークが VGG-16 の SSD

Fig. 6 SSD with VGG-16.

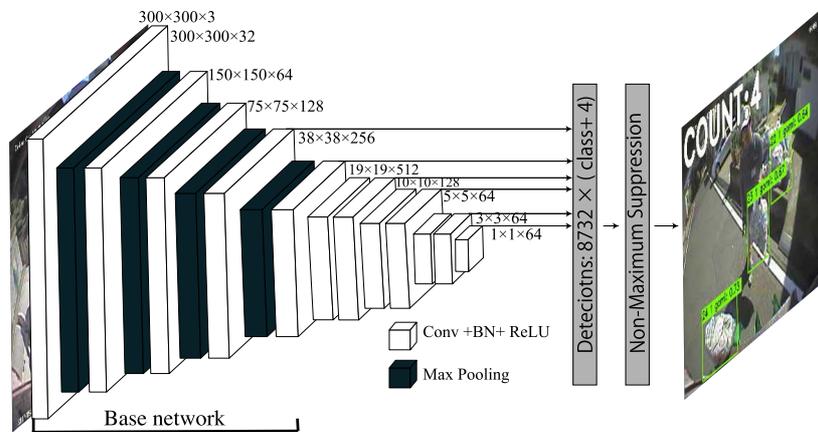


図 7 ベースネットワークが 5 層の SSD

Fig. 7 SSD with five-layers.

表 2 DTC アルゴリズムのパラメータの説明

Table 2 Table of notations.

パラメータ	説明
τ_c	検出した物体がゴミ袋である確率の閾値である。閾値以上のときに、Tracking プロセスへ送られる。
τ_{iou}	Tracking プロセスにおいてフレーム間でゴミ袋の同一性を判定するための閾値である。
τ_f	τ_f フレーム連続で Tracking しない場合、そのゴミ袋は Tracking リストから削除される。本研究では $\tau_f = 3$ を使用する。
x_{min}, y_{min}	BB の左上を表す座標。
x_{max}, y_{max}	BB の右下を表す座標。

ゴミ袋の BB を受け取ると、現在のフレームと前のフレームで検出したゴミ袋の同一性を判定する。Tracking プロセスのアルゴリズムを **Algorithm 1** に示す。フレーム間におけるゴミ袋の同一性を判定するため、DTC は Tracking の対象となっているゴミ袋のリストを保持する。このリストを TrackingList (TL) と呼ぶ。TL の各要素は検出したゴ

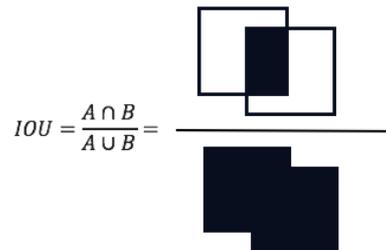


図 9 IoU の定義

Fig. 9 Definition of IoU.

ミ袋を識別するための ID と BB からなる。Tracking プロセスでは、現在のフレームにおいて検出したゴミ袋の BB 情報と、TL に保持しているすべてのゴミ袋の BB 情報を用いて IoU 値を算出する。算出した IoU 値が閾値 τ_{iou} を超えた場合、Tracking プロセスは現在のフレームで検出したゴミ袋と TL に保持しているゴミ袋で IoU 値の降順に従いマッチングを行い、同一性を判定する。検出したゴミ袋を TL 内のあるゴミ袋と同一のものとして判定した場合、同一と判定した TL のゴミ袋の BB 情報を検出したゴミ袋の

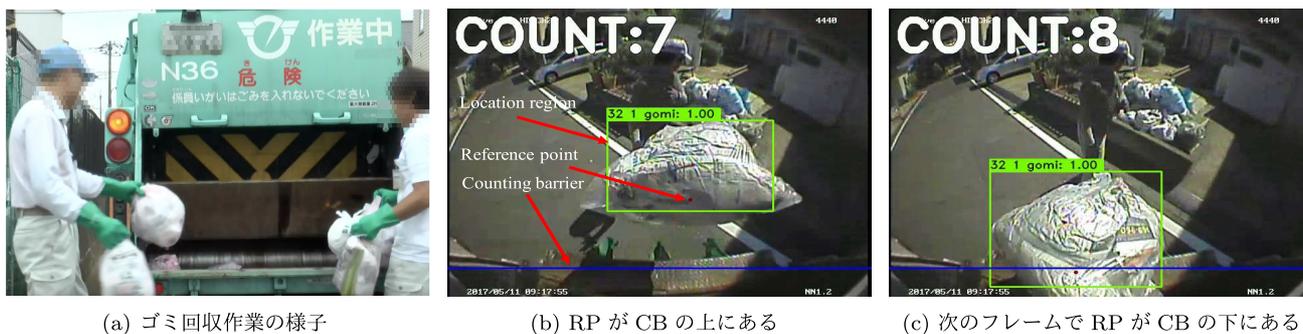


図 8 図 8(c) でゴミ袋を回収したと判定され、左上の count が 7 → 8 に増加している
Fig. 8 The reference point is below the counting barrier, so it is counted as being collected, i.e., count 7 → 8.

Algorithm 1 IoU を用いた Tracking アルゴリズム

Input: \mathcal{P}, \mathcal{C}
 $\{\mathcal{P} := \{p_0, p_1, \dots\}$ be the array of indexed objects detected in the previous frame.
 $\{\mathcal{C} := \{c_0, c_1, \dots\}$ be the array of objects detected in the current frame.
1: Initialize: $\mathcal{P}', \mathcal{C}' \leftarrow \emptyset$
2: Compute $\mathcal{O} := \{o_{c_i, p_j} \leftarrow IoU(c_i, p_j)\}, \forall c_j \in \mathcal{C}, p_i \in \mathcal{P}$
{Calculate the IoU matrix}
3: **while** $\min\{\mathcal{C}.len, \mathcal{P}.len\} \neq \mathcal{C}'.len$ **and** $\arg \max \mathcal{O} \geq \tau_{iou}$ **do**
4: $\{c^*, p^*\} \leftarrow \arg \max\{o_{c_i, p_j} : \forall c_j \in \mathcal{C}, p_i \in \mathcal{P}\}$
5: $p^*.location \leftarrow c^*.location$
6: $p^*.frameNO \leftarrow c^*.frameNO$
7: $\mathcal{P}'.add(p^*)$
8: $\mathcal{C}'.add(c^*)$
9: $o_{c_i, p_j} \leftarrow -1, \forall o_{c_i, p_j} \in \mathcal{O}$ where $p_j = p^*$ **or** $c_i = c^*$
10: **end while**{Track the detected objects}
11: **for all** $c \in \mathcal{C} - \mathcal{C}'$ **do**
12: $p.init(c.location, c.frameNO)$
13: $\mathcal{P}'.add(p)$
14: **end for**{Add new detected objects}
15: **for all** $p \in \mathcal{P} - \mathcal{P}'$ **do**
16: **if** $CurrentFrameNO - p.frameNO \leq \tau_f$ **then**
17: $\mathcal{P}'.add(p)$
18: **end if**
19: **end for**{Remove the lost objects}
Output: $\mathcal{P} \leftarrow \mathcal{P}'$ {Update the tracked objects}

BB 情報に更新する。検出したゴミ袋が TL に存在するどのゴミ袋とも一致しない場合は、新しいゴミ袋として TL に追加する。また、 τ_f フレームの間いづれのゴミ袋とも一致しなかった場合、TL に保持しているゴミ袋を TL 内から削除する。

4.3 Counting プロセス

図 8(a) のように、作業員はゴミ袋を清掃車後部にあるトランクに投入する。このゴミ回収業務のプロセスから、図 8(b) と図 8(c) に示すように、検出したゴミ袋をその位置に基づいて計数する。図 8(b) および図 8(c) で示されている Counting Barrier (CB) と呼ぶ判定線を画像の下から 1/10 の高さに設定した。清掃車と地面の境目は画像の

下から 1/10 の高さより少し高い位置にあり、CB を画像下から 1/10 の高さに設定すれば路面標識などの誤検出を減少できると考えた。また、図 8(b) で赤色で示している Reference Point (RP) と呼ぶ計数を判定する点を、BB の左上座標から横に 1/2、縦に 6/7 の位置に設定した。横から 1/2 に設定したのは、実験を行うときに RP を見やすくするためである。また、RP の縦の位置を高い位置に設定してしまうと、ゴミ袋が CB を通過して計数判定されるとき、ゴミ袋の一部が画面下へ見切れてしまい検出されない事例が見られたため、6/7 に設定した。

TL に保持しているゴミ袋は、以下の 3 つの条件を満たした場合に回収したものとして計数する。

- 前のフレームでゴミ袋の RP が CB の上にある。
- 現在のフレームでゴミ袋の RP が CB の下にある。
- ゴミ袋が 2 つのフレームにおいて同じ ID を持つ。

計数した TL にあるゴミ袋の情報は削除される。また、回収されたゴミ袋は清掃車内部へ押し込まれる前までに、動画のフレームに写り続けることがある。そのため、誤った計数を防ぐため、RP が CB の下にあるとき検出したゴミ袋の Tracking は行わない。

5. 評価

5.1 実験

実験ではゴミ回収動画を使用し、DTC の処理速度と計数精度を評価した。実験には NVIDIA GeForce GTX 1080 GPU を搭載した Desktop コンピュータと Jetson TX2 を使用した (以後、Desktop と Jetson と呼ぶ)。それぞれのプラットフォームの仕様を表 3 にまとめる。

5.1.1 データセット

SSD を学習するために、ゴミ回収動画からデータセットを作成した。8 台の清掃車が 2017 年 5 月の午前 8 時から午前 11 時の間に撮影した動画を含む。なお、動画は約 2 分ごとで分割され、動画の解像度は 640×480 で、フレームレートは約 8FPS である。動画撮影時の天候は晴れであった。



図 10 学習データセットの例

Fig. 10 Examples of the training data set. The location region of the bags are labeled manually.

表 3 実験プラットフォームの詳細

Table 3 Specifications of the experiment platforms.

	Desktop	Jetson TX2
CPU	Intel® Core™ i7-6950X, 3.00 GHz, 10 Cores	Dual Denver 2/2 MB L2, 2 GHz, 2 Cores+ Quad ARM®A57/2 MB L2, 2 GHz, 4 Cores
GPU	GeForce GTX1080, 1,733 MHz, 2560 CUDA Cores	NVIDIA Pascal, 1,300 MHz, 256 CUDA Cores
Memory	GDDR5X, 256 bit, 8 GB, 320 GB/s	LPDDR4, 128 bit, 8 GB, 59.7 GB/s
OS	Ubuntu 16.04	Ubuntu 16.04
Software	Python 3.7.0, PyTorch 0.41, OpenCV 3.4.3	Python 3.7.0, PyTorch 0.41, OpenCV 3.4.3

表 4 5層のベースネットワークのアーキテクチャ

Table 4 Five-layer base network architecture.

Type	Filters	Size	Output
Convolutional	32	3 × 3	300 × 300
Max Pooling	-	2 × 2	150 × 150
Convolutional	64	3 × 3	150 × 150
Max Pooling	-	2 × 2	75 × 75
Convolutional	128	3 × 3	75 × 75
Max Pooling	-	2 × 2	38 × 38
Convolutional	256	3 × 3	38 × 38
Max Pooling	-	2 × 2	19 × 19
Convolutional	512	3 × 3	19 × 19

図 1(a) で示したように、ゴミ排出量全体において可燃ゴミが大部分を占めていることから、可燃ゴミ袋の排出量を取得できればゴミ全体の排出量の大部分を得ることができる。そのため、本研究では可燃ゴミを評価の対象とした。8つの動画のうち、2つを物体検出モデルの学習用データセット作成に使用し、残る6つを評価用データセット作成と評価実験に使用した。

学習用の2つの動画をすべてフレーム化し、そのうちゴミ袋が写っているフレームのみを抽出し、図 10のように、ゴミ袋を矩形でアノテーションし、約5,500個の学習用データセットを作成した。すべての評価実験で、この学習用データセットで学習したモデルを使用した。

同様の手順で、6つの動画からそれぞれ約80個ずつ、合計約500個の評価用データセットを作成した。

5.1.2 学習

実験では、ベースネットワークとして VGG-16, Mobilenet v2 [30], VGG-16 を参考にベースネットワークの層を 6, 5, 4, 3, 2 層に削減した合計7つの SSD を学習した。層を削減したモデルは、Convolutional 層の後に Batch Normalization [31] による正規化を行い、Max Pooling で down sampling を繰り返す VGG-16 の構造を参考にした。

また、層の削減は Convolutional 層の削減である。

一例として、5層に削減したベースネットワークの構造を表 4 に示す。活性化関数には ReLU [32] を使用した。Mobilenet v2 はモバイルデバイス向けに開発された Mobilenet [33] を改良した、さらに速い処理速度と高い精度を持つ CNN アーキテクチャである。本研究では、SSD のベースネットとして使用するため一部 Convolutional 層のストライドを 2 から 1 へ変更した Mobilenet v2 を使用している。7つの SSD すべて同じ条件で学習を行った。重みの初期値として、xavier [34] の初期値を使用した。また、入力画像は 640 × 480 から 300 × 300 にリサイズした。そのため、処理速度は向上する一方、検出精度は低下する。学習は、500 epoch 行い、学習率の初期値は $1e-3$ であり、350 epoch 後に $1e-4$ に、450 epoch 後に $1e-5$ へと減少させた。その他のハイパーパラメータは文献 [21] に従う。

5.2 結果

実験では以下の5つの項目について評価を行った。

- Detection プロセスにおけるゴミ袋の検出精度
- DTC の処理速度
- Detection プロセスにおける最適な閾値 τ_c の選択
- Tracking プロセスにおける最適な閾値 τ_{iou} の選択
- 異なる6台の清掃車における DTC の計数精度

表 6 Pascal VOC における Average Precision
Table 6 Average Precision in Pascal VOC dataset.

VOC 2007 test	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
VGG-16	74.2	79.3	83.7	68.7	65.6	46.8	83.7	86.0	85.0	57.0	75.0	75.5	78.7	85.1	84.6	77.2	44.9	72.2	76.6	84.5	73.6
5層	56.2	64.1	66.7	42.2	41.3	21.4	69.0	73.4	69.9	35.7	53.2	54.7	62.3	73.5	68.7	61.8	25.2	55.5	58.0	70.2	57.3
Mobilenetv2	54.6	61.9	62.7	43.0	40.7	17.6	69.5	62.8	69.1	39.6	43.3	65.1	59.9	70.9	70.2	53.3	22.3	49.3	67.3	71.3	51.9

表 5 異なるベースネットワークの Average Precision
Table 5 Average Precision of different base networks.

	VGG-16	Mobile v2	6層	5層	4層	3層	2層
AP	79.8	78.8	77.9	78.9	50.6	33.6	9.1

5.2.1 Detection プロセスにおけるゴミ袋の検出精度

本研究では, Detection プロセスにおけるゴミ袋の検出精度として, 文献 [14] の定義に基づいた, Average Precision (AP) の算出方法を採用した. AP は約 500 個の評価用データセットを使用して算出した.

AP の評価結果を表 5 に示す. 表 5 から, ベースネットワークのなかで VGG-16 の AP が最も高いが, Mobilenet v2, 6層, 5層の AP と大差はないことが分かる. 一方, 4層になると AP が 50.6 と大きく精度が低下し, 2層になると 9.1 となりほとんどゴミ袋を検出できていない.

追加実験として, ゴミ袋の検出精度で AP が高かった VGG-16, 5層, Mobilenet v2 のそれぞれをベースネットワークとする SSD に Pascal VOC [14] のデータセットを学習させ, mAP を測定した. 表 6 には, Pascal VOC に含まれる aero, bike など 20 の物体の AP と 20 の物体の AP について平均した mAP を示す. 表 6 から, VGG-16 がベースネットワークの SSD の mAP が最も高いことが分かる. ゴミ袋検出の AP では 3 つのベースネットワークでそれほど差が出なかった一方, Pascal VOC データで大きく差が出たのは, ゴミ袋検出の場合はクラスが 1 種類の比較的簡単な検出であり, 少ない層でもゴミ袋の特徴量を獲得できたためだと考えられる. 一方で, Pascal VOC では, 20 クラスの物体の特徴量を少ない層ですべて獲得することができず, 層の深浅で mAP に大きく差が出たと考えられる.

5.2.2 DTC の処理速度

評価用動画 6 つのうちから 2 分間のゴミ回収動画を用いて, desktop と Jetson 両プラットフォームで DTC の処理速度を測定した. 指標は FPS を用いた.

各プラットフォームで 5 回処理速度を測定し, その平均した FPS 値を表 7 にまとめた. 表 7 から, 層が減っていくに従って処理速度が上昇していることが分かる. 特にベースネットワークが VGG-16 の SSD 以外は 8FPS を超えており, 走行中の清掃車において, リアルタイムでのゴミ袋を計数することが可能である. 以上の実験結果から, リアルタイム処理が可能なのなかで最も AP が高かった 5 層からなるベースネットワークの SSD を, DTC の検

表 7 異なるベースネットワークの FPS

Table 7 FPS in different base networks.

	VGG-16	Mobile v2	6層	5層	4層	3層	2層
Jetson TX2	5.2 fps	11.2 fps	14.0 fps	15.2 fps	16.6 fps	18.1 fps	18.7 fps

出器として採用した.

5.2.3 Detection プロセスにおける最適な閾値 τ_c の選択

SSD により物体を検出する際, BB の座標とともに検出した物体がゴミ袋である確率も出力する. Detection プロセスでは, この確率が閾値 τ_c 以上である場合のみ, 検出した物体をゴミ袋であると判定し, その BB を Tracking プロセスへ送る. DTC におけるゴミ袋計数の精度を最大化する最適な閾値 τ_c を探索するため, 異なる τ_c が計数の精度に与える影響を評価した. 評価用動画 6 つのうちから合計 30 分間のゴミ回収動画を用いて, 以下のように行った.

- (1) DTC が計数したゴミ袋の個数を記録.
- (2) 人が動画を見て計数した個数を記録.
- (3) 得られた数値をもとに, Recall, Precision, F-measure を算出.

Recall, Precision, F-measure をそれぞれ式 (1)–(3) に示す. Recall はすべての回収されたゴミ袋のうち, DTC が正しく計数できたゴミ袋の個数の割合で, Precision は DTC がゴミ袋として計数した個数のうち, 実際にゴミ袋であった個数の割合, F-measure は Recall と Precision の調平均である. ここで, TP は DTC が正しく計数できたゴミ袋の個数, FN は DTC が見逃して計数できなかったゴミ袋の個数, FP は DTC がゴミ袋ではないものを誤ってゴミ袋として計数した個数である. たとえば, DTC が 3 つのゴミ袋を回収したと推定した一方, 人は 4 つのゴミ袋が回収されたと推定したとき, $TP = 3$, $FN = 1$ となる.

$$Recall = \frac{\text{No. of TPs}}{\text{No. of TPs} + \text{No. of FNs}} \quad (1)$$

$$Precision = \frac{\text{No. of TPs}}{\text{No. of TPs} + \text{No. of FPs}} \quad (2)$$

$$F - measure = \frac{2Recall \times Precision}{Recall + Precision} \quad (3)$$

上述した実験プロセスに従い, 異なる τ_c において Recall, Precision, F-measure の指標に基づき, DTC のゴミ袋の計数精度を評価した. このとき, Tracking プロセスにおける閾値 τ_{iou} を 0 に設定した.

評価結果を表 8 に示す. 表 8 より, τ_c が 0.05 から 0.8 に増加するにつれて, Precision の値は 0.34 から 0.90 に増加

表 8 $\tau_{iou} = 0$ における異なる τ_c での DTC の計数精度

Table 8 Accuracy of DTC Vs. τ_c where $\tau_{iou} = 0$.

τ_c	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.8
Recall	0.68	0.66	0.54	0.55	0.52	0.49	0.44	0.29
Precision	0.34	0.33	0.71	0.84	0.91	0.92	0.92	0.90
F-measure	0.45	0.44	0.61	0.66	0.66	0.64	0.60	0.44

表 9 $\tau_c = 0.4$ における異なる τ_{iou} での DTC の計数精度

Table 9 Accuracy of DTC Vs. τ_{iou} where $\tau_c = 0.4$.

τ_{iou}	0	0.001	0.005	0.01	0.05	0.1
Recall	0.52	0.51	0.51	0.51	0.42	0.37
Precision	0.91	0.91	0.91	0.91	0.89	0.84
F-measure	0.66	0.65	0.65	0.65	0.58	0.52

するが、Recall は 0.68 から 0.29 に減少することが分かる。また、F-Measure は最初増加し、次いで減少している。これらの結果は、閾値 τ_c を上げることで、ゴミ袋でないものを誤って計数することを低減できる一方、同時にゴミ袋を正しく計数する精度を低下させること示す。実験結果に基づき、以後の評価では F-Measure が最も高かった $\tau_c = 0.4$ を採用した。

5.2.4 Tracking プロセスにおける最適な閾値 τ_{iou} の選択

本項では、Tracking プロセスにおける IoU の閾値 τ_{iou} が DTC の計数精度に及ぼす影響を評価する。実験のプロセスは閾値 τ_c を求めるプロセスと同様である。

表 9 に、閾値 τ_{iou} が 0 から 0.1 まで変化するときの DTC の計数精度を示す。ここで、閾値は $\tau_c = 0.4$ である。表 9 から、 τ_{iou} が増加すると、Recall, Precision, F-measure それぞれ 3 つの評価指標はすべて単調に減少していくことが分かる。これは実験で使用している動画のフレームレートが 8FPS と低いことが原因であると考えられる。動画のフレームレートが低いため、連続する 2 つのフレーム間においてゴミ袋の移動距離が大きくなり、ゴミ袋の重なる位置領域が小さくなる。そのため、閾値 τ_{iou} を高く設定すると τ_{iou} を満たすことができず、Tracking に失敗する。したがって、 τ_{iou} の増加によって 3 つすべての指標が単調に減少していく結果となった。実験結果より、F-measure が最も高かった $\tau_{iou} = 0$ を選択した。

5.2.5 異なる 6 台の清掃車における計数精度評価

最後に、ドライブレコーダの設置位置や角度、作業員など変動要素が存在する環境における DTC の計数精度を検証するため、6 つの評価用動画すべてを用いて実験を行った。各動画について Recall, Precision, F-measure を算出することで、異なる環境下での影響を調べる。

実験結果を表 10 と表 11 にまとめる。閾値はそれぞれ $\tau_c = 0.4$, $\tau_{iou} = 0$ に設定した。表 10 から、平均 Recall は 0.67, 平均 Precision は 0.82 であることが分かった。平均 Recall が 0.67 であることは清掃車が回収した全ゴミ袋のうち 67% を計数できた一方で、33% を見逃して計数でき

表 10 異なる 6 台の清掃車における DTC の計数精度

Table 10 Accuracy of DTC Vs. Trucks.

Truck	No.1	No.2	No.3	No.4	No.5	No.6	Average
Recall	0.64	0.66	0.69	0.61	0.69	0.72	0.67
Precision	0.92	0.82	0.84	0.70	0.88	0.81	0.82
F-measure	0.76	0.73	0.76	0.65	0.77	0.76	0.74

表 11 異なる 6 台の清掃車における DTC の TP, FN, FP

Table 11 TP, FN and FP of DTC Vs. Trucks.

Truck	No.1	No.2	No.3	No.4	No.5	No.6	Sum
TP	537	538	550	494	514	522	3155
FN	300	277	245	311	235	201	1569
FP	44	121	103	214	70	126	678



図 12 作業員の帽子を誤って計数した例

Fig. 12 Example of counting workers' hats.

なかったことを示している。また、平均 Precision が 0.82 であることは、DTC がゴミ袋として計数したもののうち、82% はゴミ袋で、18% はゴミ袋ではないものを誤ってゴミ袋として計数してしまったことを示している。また、表 11 から、No.4 と No.1 の清掃車では FP に 4 倍の差があることが分かる。No.4 の動画を調べてみると、図 12 のように、作業員がゴミ袋を押し込むとき、帽子を連続的に誤ってゴミと計数していた。作業員が DTC の安定性に影響を与える一方、ドライブレコーダの設置位置や角度の影響は見られなかった。

5.3 考察と今後の課題

計数の Recall 低下の大きな要因として、ゴミ回収動画の FPS が 8FPS と低いことがあげられる。FPS が低いためフレーム間でのゴミ袋の画面内における移動距離が大きくなる。そのため、図 13 (左) のように、フレーム間でのゴミ袋の移動距離が小さいと領域の重なりが大きくなり、閾値 τ_{iou} を満たし、ゴミ袋を計数できる。一方で、図 13 (右) のように、フレーム間でのゴミ袋の移動距離が大きくと領域がまったく重ならず、閾値 τ_{iou} を満たせず、Detection プロセスでゴミ袋の検出は正しくできていても Tracking できず、計数に失敗する。動画の FPS を増加することで

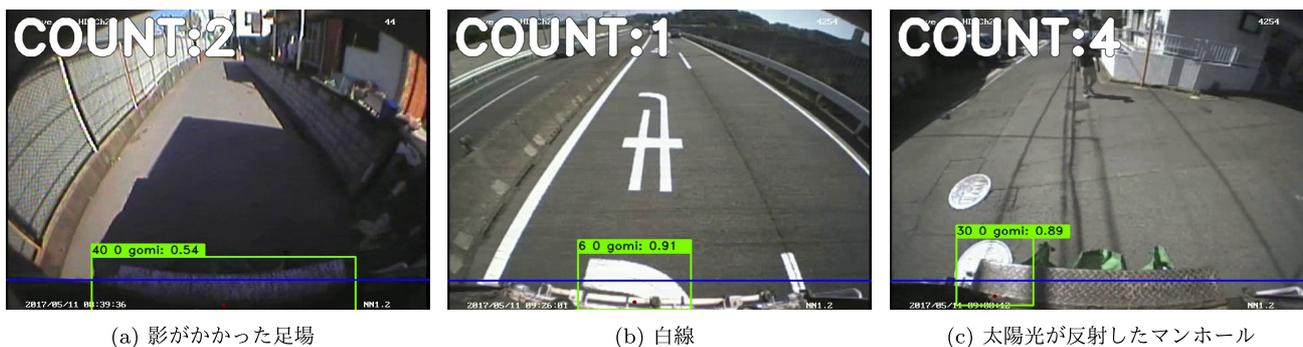


図 11 誤って異なる物体をゴミ袋として計数した例

Fig. 11 Example of counting different objects as garbage bags.



図 13 左の画像ではフレーム間でのゴミ袋の移動距離が小さいため計数に成功している一方、右の画像では移動距離が大きいため失敗している

Fig. 13 In the image on the left, counting is successful because the moving distance of the trash bag between the frames is small, while in the image on the right, it is failing because the moving distance is large.

Recall を改善できる一方、処理速度の面で新たな課題となる。そのため、DeepSORT [35] のように IoU だけに頼らない Tracking 手法を今後検討する。

また、Precision 低下の要因は、図 11 に示すように、Detection プロセスにおいて、ゴミ袋と似た特徴を持つ物体を誤って検出してしまうためである。特に図 12 の誤った計数の例では、ゴミ袋を押し込むとき、作業員の帽子が計数判定線である CB を前後するため、連続的な誤った計数が引き起こされた。これは、帽子などを新しいクラスとして分類し検出することで低減できると考えられる。

今後実用化するためには精度の改善が求められる。今回の実験では、作業員の帽子を除き環境的差異の影響は見られなかった。つまり、帽子が要因となるもの以外、FN と FP は一様に分布して発生すると考えられる。したがって、帽子をクラス分類し、藤沢市全体のゴミ排出量データと比較することで、実用可能な各地点のゴミ排出量データの推定が可能になると考える。そのための具体的な目標とし

て、Recall と Precision それぞれ 0.80, 0.90 を目指し改善していく。

今回は可燃ゴミ 1 クラスと比較的簡単な物体検出タスクであったため、ある一定以上の深さを持つベースネットワークにおいてはその深浅によって検出精度の指標である AP に差がそれほどなかった。一方、Pascal VOC のように多クラスの物体検出タスクだと、深浅によって大きく差が現れた。そのため、可燃ゴミだけでなく不燃ゴミやプラスチックなど多種類のゴミ袋を検出するには、RefineDet [27] や M2Det [36] のように SSD より検出精度が高い物体検出手法が求められる。一方、これらは SSD と比較すると処理速度の面で劣るため、より処理速度の高速化が求められる。

図 11 (a) や図 11 (c) のように、影や太陽光など、明度が精度に影響を与える事例がみられた。今回学習したデータセットは晴れのみであるため、学習したデータセットと明度が異なる曇りや雨、夜間など暗い環境下では、精度の低下が考えられる。そのため、異なる環境のデータセット作成と、明度が精度にもたらす影響の検証実験を検討する。

今後はこれらの点を改善していくとともに、DeepCounter によって得られたゴミの排出量データを可視化し住民へフィードバックするアプリケーションの開発を検討している。また、排出量データを分析し、いつ、どこでゴミの排出量が増減するのか、つまり時空間的なゴミ排出量変化の傾向を明らかにし、清掃車のルート最適化など、効率的都市の実現に役立つ手法を検討する。回収作業員の方に話をうかがったところ、ゴミ袋の大きさの分類や重さの測定ができるとう助かるといった意見があった。そのため、今後ゴミ袋の大きさを分類する手法と、大きさからゴミ袋の重さを推定する手法の開発も検討する。

6. 結論

深層学習による物体検出技術と清掃車に搭載可能な組み込みシステムを組み合わせることで、回収したゴミ袋の個数をセンシングするシステム、DeepCounter を提案した。DTC アルゴリズムを実装し、ゴミ回収動画を用いて評価実験を行った。実験では、システムの有効性が示された一

方、実用的なシステムとして用いるには、考察で述べた問題点を改善する必要がある。今後、考察で述べた問題点を改善していくとともに、DeepCounterによって得られたゴミ排出量データを実際に利活用していく方法を検討する。本研究の提案手法によって得られるデータを利活用することで、清掃車のルート最適化や排出量削減のための啓蒙など、様々な行政施策に寄与することが期待される。

謝辞 本研究の一部は科学研究費補助金(19H04099)の助成を受けたものである。

参考文献

- [1] Kitagawa Amagasa Data Engineering Laboratory, Tsukuba University: Gomi G1 Grand Prix, available from (<http://bigdata.kde.cs.tsukuba.ac.jp/~g1gp/index.html>).
- [2] Mikami, K., Chen, Y., Nakazawa, J., Iida, Y., Kishimoto, Y. and Oya, Y.: DeepCounter: Using Deep Learning to Count Garbage Bags, *2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pp.1–10 (online), DOI: 10.1109/RTCSA.2018.00010 (2018).
- [3] Chen, Y., Yonezawa, T., Nakazawa, J. and Tokuda, H.: Evaluating the Spatio-temporal Coverage of Automotive Sensing for Smart Cities, *10th International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2017)* (2017).
- [4] Rezaei, S. and Sengupta, R.: Kalman Filter-Based Integration of DGPS and Vehicle Sensors for Localization, *IEEE Trans. Control Systems Technology*, Vol.15, No.6, pp.1080–1088 (online), DOI: 10.1109/TCST.2006.886439 (2007).
- [5] Fujii, S., Fujita, A., Umedu, T., Kaneda, S., Yamaguchi, H., Higashino, T. and Takai, M.: Cooperative Vehicle Positioning via V2V Communications and Onboard Sensors, *2011 IEEE Vehicular Technology Conference (VTC Fall)*, pp.1–5 (online), DOI: 10.1109/VETECE.2011.6093218 (2011).
- [6] Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S. and Balakrishnan, H.: The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring, *Proc. 6th International Conference on Mobile Systems, Applications, and Services, MobiSys '08*, pp.29–39, ACM (online), DOI: 10.1145/1378600.1378605 (2008).
- [7] Kawano, M., Mikami, K., Yokoyama, S., Yonezawa, T. and Nakazawa, J.: Road marking blur detection with drive recorder, *2017 IEEE International Conference on Big Data*, pp.4092–4097 (online), DOI: 10.1109/BigData.2017.8258427 (2017).
- [8] Du, R., Chen, C., Yang, B., Lu, N., Guan, X. and Shen, X.: Effective Urban Traffic Monitoring by Vehicular Sensor Networks, *IEEE Trans. Vehicular Technology*, Vol.64, No.1, pp.273–286 (online), DOI: 10.1109/TVT.2014.2321010 (2015).
- [9] Hasenfrazz, D., Saukh, O., Walser, C., Hueglin, C., Fierz, M. and Thiele, L.: Revealing the Limits of Spatio-temporal High-resolution Pollution Maps, *Proc. 11th ACM Conference on Embedded Networked Sensor Systems, SenSys '13*, pp.28:1–28:2, ACM (online), DOI: 10.1145/2517351.2517384 (2013).
- [10] Chen, Y., Nakazawa, J., Yonezawa, T., Kawasaki, T. and Tokuda, H.: An Empirical Study on Coverage-Ensured Automotive Sensing Using Door-to-door Garbage Collecting Trucks, *Proc. 2nd International Workshop on Smart, SmartCities '16*, pp.6:1–6:6, ACM (online), DOI: 10.1145/3009912.3009921 (2016).
- [11] Kishino, Y., Shirai, Y., Takeuchi, K., Naya, F., Ueda, N., Chen, Y., Yonezawa, T. and Nakazawa, J.: Detecting Garbage Collection Duration Using Motion Sensors Mounted on Garbage Trucks Toward SmartWaste Management, *SPWID 2017: 3rd International Conference on Smart Portable, Wearable, Implantable and Disability-oriented Devices and Systems* (2017).
- [12] Viola, P. and Jones, M.: Rapid object detection using a boosted cascade of simple features, *Proc. 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001*, Vol.1, p.I, IEEE (2001).
- [13] Dalal, N. and Triggs, B.: Histograms of oriented gradients for human detection, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, CVPR 2005*, Vol.1, pp.886–893, IEEE (2005).
- [14] Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J. and Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results (2012).
- [15] Zhao, Z.-Q., Zheng, P., Xu, S.-T. and Wu, X.: Object detection with deep learning: A review, arXiv preprint arXiv:1807.05511 (2018).
- [16] Uijlings, J.R., Van De Sande, K.E., Gevers, T. and Smeulders, A.W.: Selective search for object recognition, *International Journal of Computer Vision*, Vol.104, No.2, pp.154–171 (2013).
- [17] Girshick, R., Donahue, J., Darrell, T. and Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.580–587 (2014).
- [18] Girshick, R.: Fast r-cnn, *Proc. IEEE International Conference on Computer Vision*, pp.1440–1448 (2015).
- [19] Ren, S., He, K., Girshick, R. and Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks, *Advances in Neural Information Processing Systems*, pp.91–99 (2015).
- [20] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A.: You only look once: Unified, real-time object detection, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.779–788 (2016).
- [21] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. and Berg, A.C.: Ssd: Single shot multi-box detector, *European Conference on Computer Vision*, pp.21–37, Springer (2016).
- [22] Erhan, D., Szegedy, C., Toshev, A. and Anguelov, D.: Scalable object detection using deep neural networks, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.2147–2154 (2014).
- [23] Redmon, J. and Farhadi, A.: YOLO9000: Better, Faster, Stronger, arXiv preprint arXiv:1612.08242 (2016).
- [24] Redmon, J. and Farhadi, A.: YOLOv3: An Incremental Improvement, arXiv (online), available from (<https://arxiv.org/abs/1804.02767>) (2018).
- [25] Lin, T.-Y., Goyal, P., Girshick, R., He, K. and Dollár, P.: Focal loss for dense object detection, arXiv preprint arXiv:1708.02002 (2017).
- [26] Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A. and Berg, A.C.: DSSD: Deconvolutional single shot detector, arXiv preprint arXiv:1701.06659 (2017).
- [27] Zhang, S., Wen, L., Bian, X., Lei, Z. and Li, S.Z.:

- Single-shot refinement neural network for object detection, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.4203-4212 (2018).
- [28] Simonyan, K. and Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition, *ArXiv e-prints* (online), available from <https://arxiv.org/abs/1409.1556> (2014).
- [29] Fukunaga, K. and Hostetler, L.D.: The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition, *IEEE Trans. Information Theory*, Vol.21, No.1, pp.32-40 (online), DOI: 10.1109/TIT.1975.1055330 (1975).
- [30] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.-C.: Mobilenetv2: Inverted residuals and linear bottlenecks, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.4510-4520 (2018).
- [31] Ioffe, S. and Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167 (2015).
- [32] Nair, V. and Hinton, G.E.: Rectified linear units improve restricted boltzmann machines, *Proc. 27th International Conference on Machine Learning (ICML-10)*, pp.807-814 (2010).
- [33] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861 (2017).
- [34] Glorot, X. and Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks, *Proc. 13th International Conference on Artificial Intelligence and Statistics*, pp.249-256 (2010).
- [35] Wojke, N., Bewley, A. and Paulus, D.: Simple Online and Realtime Tracking with a Deep Association Metric, *2017 IEEE International Conference on Image Processing (ICIP)*, pp.3645-3649, IEEE (online), DOI: 10.1109/ICIP.2017.8296962 (2017).
- [36] Zhao, Q., Sheng, T., Wang, Y., Tang, Z., Chen, Y., Cai, L. and Lin, H.: M2Det: A Single-Shot Object Detector based on Multi-Level Feature Pyramid Network, *33rd AAAI Conference on Artificial Intelligence, AAAI* (2019).



三上 量弘

2019年慶應義塾大学環境情報学部環境情報学科卒業。現在、同大学大学院政策・メディア研究科修士課程在学中。主に、スマートシティ、深層学習、ユビキタスコンピューティング等の研究に従事。



陳 寅 (正会員)

2008年中国西安電子科技大学計算機科学部卒業。2011年同大学大学院計算機システム研究科修士。2014年公立はこだて未来大学博士号取得(システム情報学)。2014年10月まで同大学博士研究員。2014年慶應義塾大学大学院政策・メディア研究科特任助教。現在、同大学院政策・メディア研究科特任講師。主に、Mobile network, Sensor networkなどの研究に従事。IEEE, ACM, IPSJ各会員。



中澤 仁 (正会員)

慶應義塾大学環境情報学部教授。博士(政策・メディア)。1975年生。1998年慶應義塾大学総合政策学部卒業。2001年同大学大学院政策・メディア研究科修士課程修了。2001年同大学院同研究科博士課程修了。ミドルウェア, システムソフトウェア, ユビキタスコンピューティング, センサネットワーク等の研究に従事。電子情報通信学会, ACM, IEEE各会員。