

窓の反射を考慮したバスの乗降者人数の計数手法に関する研究 A Study on Counting Method of the Number of Passengers on Bus Considering Window Reflection

松井 友哉[†] 中原 匡哉[‡] 佐伯 祐介[‡]
Yuya Matsui Masaya Nakahara Yusuke Saeki

1. はじめに

現在、バス業界では、利用者の減少や運転手不足などの影響[1]により路線の見直しが進められている。また、こうした路線の見直しはバス会社のみならず、バス会社に依頼してバスを運行している大学などの施設でも進められている。この路線の見直しの際には、バスの利用者数を把握する必要があり、施設などでは主に人手で目視により計数されている。しかし、人手による計数では、人的コストや人的被害の観点から自動で計数する技術の確立が求められている。既存研究[2][3]では、バス車内にビデオカメラやセンサーなどを設置して、乗客を計数する手法が多く提案されている。しかし、これらの手法では、乗客人数のデータを取得するために、施設がバス会社に対して多くの金銭的コストを支払う必要がある課題や、バス車内が混雑した際に誤計数する可能性がある。これらの課題を解決するため、既存研究[4]では、バス車外から利用者数を調査する手法を提案した。しかし、本手法では、バスの降車人数を計数する際に、窓の反射により誤計数する課題がある。そこで、本研究では、既存研究[4]のバスの降車人数を計数する手法を基に、深層学習を用いてバスの窓を認識することで誤計数を抑制する手法を提案する。また、バスの乗車人数を計数する手法も新たに提案する。

2. 研究概要

提案手法の処理フローを図 1 に示す。提案手法は、降車人数計数機能と乗車人数計数機能により構成される。入力データは、ビデオカメラで撮影した動画像、出力データは、バスから降車した人数とバスに乗車する人数である。本研究で新たに作成したのは、降車人数計数機能のバスの窓を塗りつぶす処理と乗車人数計数機能である。

2.1 降車人数計数機能

本機能では、深層学習を用いてバスの車体、バスの窓、バス降車口と人物を検出し、それらを基にバスから降車する人数を計数する。本機能は、窓塗りつぶし処理、人物追跡処理、降車人数計数処理により構成される。

窓塗りつぶし処理では、動画像からバスを検出し、窓を塗りつぶして、人物の検出範囲を自動で決定する。まず、動画像からバスを検出する。次に、計数対象のバスを選定する。このとき、検出したバスの中で、矩形面積が最も大きいバスをビデオカメラに最も近い位置にあるバスとして

[†] 大阪電気通信大学大学院 総合情報学研究科

Graduate School of Osaka Electro-Communication University,
Graduate School of Information Science and Arts, Osaka 575-0063, Japan

[‡] 大阪電気通信大学 総合情報学部

Osaka Electro-Communication University,
Faculty of Information Science and Arts, Osaka 575-0063, Japan

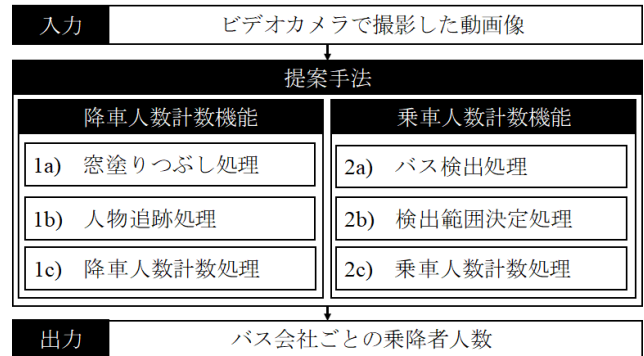


図 1 提案手法の処理フロー

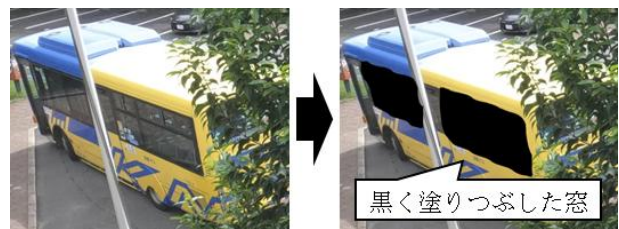


図 2 バスの窓を塗りつぶした例

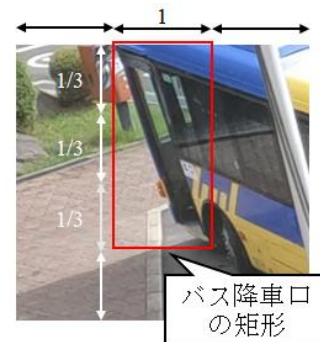


図 3 人物の検出範囲

推定し、選定する。次に、選定したバスの降車口を検出した矩形がフレーム間で一定の割合以上重なっている場合、バスは停車していると推定する。次に、事前にバス停に停車するバスの車体色を基に設定した色の閾値から、選定したバスの会社を推定する。そして、選定したバスの窓を検出して、図 2 に示すように黒く塗りつぶす。これにより、バスの窓の反射による誤計数の課題を解消できる。最後に、図 3 に示すようにバスの降車口付近を人物の検出範囲に自動で決定する。これは、通行人などの影響による誤計数を防止するためである。本処理におけるバスの降車口を検出する学習モデルは、既存研究[4]と同様のものを用いる。本

学習モデルの構築には、YOLO[5]の後継であるYOLOv8[6]が用いられている。本研究では、バスの窓を検出する学習モデルの構築や人物の検出には、YOLOv8の後継であるYOLO11[7]を用いることとした。

人物追跡処理では、窓塗りつぶし処理で決定した人物の検出範囲から人物を検出し、追跡する。まず、YOLO11の既存の学習モデルを用いて人物を検出する。このとき、図4に示すように同一人物を複数回検出する場合がある。その際、検出した矩形同士の重なりがどちらかの一定割合以上であれば同一人物とし、各矩形の座標値を基に新たに1つの検出矩形を作成する。次に、図5に示す肩の骨格構造の座標値を基に人物を追跡する。肩の座標値を用いる理由は、矩形の重心点の座標値を用いる場合に比べて、フレーム間での大幅な変化が少なく、追跡が途切れにくいからである。ただし、肩の座標値を取得できない場合には、矩形の重心点の座標値を用いる。さらに、追跡の際には、フレーム間で肩、もしくは矩形の重心点の座標値が一定の閾値内であれば同一人物として追跡する。最後に、人物の検出漏れを考慮し、一定フレームの間に追跡されなかった場合に処理を終了する。

降車人数計数処理では、次の3つの判定条件を基に、追跡した人物がバスを降車したかどうか推定して計上する。

1つ目の条件は、図5に示す膝の骨格構造が一定の信頼度で検出できるかどうかである。これにより、誤検出結果の誤計数を抑制する。2つ目の条件は、図6に示すように人物の矩形に設定した3点それぞれをフレーム間で結んだ各線分のいずれかが、バス降車口の矩形下部に自動で決定した計数線を跨ぐかどうかである。このとき、人物の追跡開始時の矩形位置によっては計数線を跨げない可能性があるため、人物の矩形に設定する3点は、追跡開始時と追跡中で位置をずらしている。3つ目の条件は、図7に示すように人物に設定した3点とバス降車口の矩形を基に、人物がバス降車口の矩形内にあるかどうかである。バス降車口付近の通行人は追跡開始時の位置次第でバスから降りてきた乗客であると誤判定される可能性があるため、人物に設定する3点は人物の矩形中央寄りに設定することで、誤判定の可能性を低減している。本処理は一定時間内にバスを降車する乗客を計上できなかった場合、乗客の降車が全て完了したとして推定し、計数した降車人数とそのバス会社を出力する。

2.2 乗車人数計数機能

本機能では、深層学習を用いて、バス停に並んでいる人物を検出し、バスに乗車した人数を計数する。本機能は、バス検出処理、検出範囲決定処理と乗車人数計数処理により構成される。

バス検出処理では、画像内からバスが停車する範囲を設定し、動画内から範囲内に停車するバスを選定する。まず、図8に示すように画像内からバスが停車する範囲をGUI操作で設定する。このとき、バスが停車する範囲は、2点を結ぶ直線を複数設定して表現している。次に、YOLO11の既存の学習モデルを用いてバスを検出する。次に、図9に示すように設定したバスの停車範囲と検出したバスの矩形を基に、バスが停車範囲内かどうかを判定する。このとき、バスが停車範囲内であった場合、バスが停車範囲に入る直前のフレーム画像（以降、バス停車前の画像）

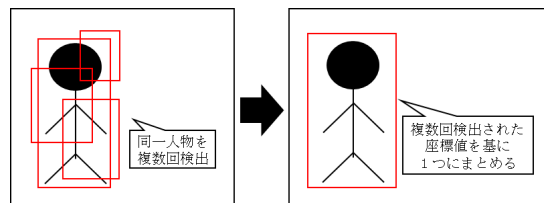


図4 同一人物を複数回検出した矩形の調整



図5 人物の骨格構造の検出

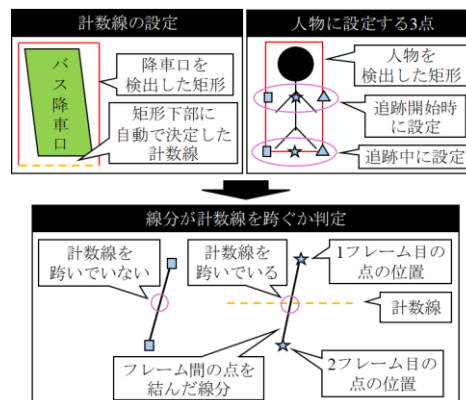


図6 計数線を跨ぐ判定

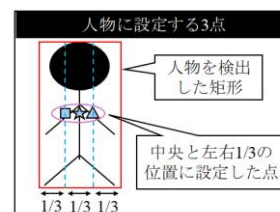


図7 バス降車口の矩形の内外判定

を保持する。これは、後の乗車人数計数処理で用いるためである。そして、検出したバスの矩形の重心点の座標値を基に停車範囲内のバスを追跡し、フレーム間で重心点の座標値が一定の移動量以下であれば、フレーム間で検出したバスは同一かつ停車していると推定し、計数対象として選



図 8 バスの停車範囲の設定例

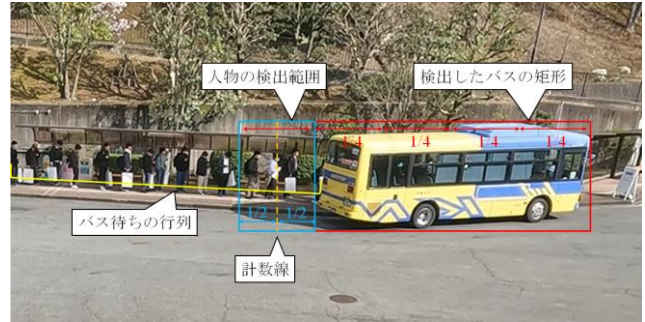


図 10 人物の検出範囲と計数線の決定



図 9 バスの停車範囲の内外判定

定する。最後に、事前にバス停に停車するバスの車体色を基に設定した色の閾値から、選定したバスの会社を推定する。

検出範囲決定処理では、人物の検出範囲と計数時に用いる計数線を自動で決定する。まず、図 10 に示すようにバスの車体後方を人物の検出範囲として自動で決定する。これは、図 10 のようにバスの乗車待ちの乗客がバスの車体後方に向けて行列を作るためである。次に、図 10 に示すように検出範囲の横軸中央に縦の計数線を自動で決定する。

乗車人数計数処理では、YOLO11 の既存の学習モデルを用いて人物を検出し、バスに乗車した人数を推定して計上する。まず、図 11 に示すように、バス停車前の画像から停車したバスの矩形範囲内の人物を検出して計上する。これは、停車したバスの陰に隠れた乗客を計上するためである。次に、図 12 に示すように、検出範囲決定処理で決定した範囲内から人物を検出し、計数線とバスの間で検出した人物を計上する。これは、乗客がバスに向かって歩くため、本処理開始時点で図中 A の範囲にいる人物が計数線を跨ぐことができず、計数漏れになるためである。次に、決定した範囲内の人物を検出して追跡する。人物の追跡は、検出した矩形の重心点の座標値がフレーム間で一定の移動量以内であれば同一人物として追跡する。そして、フレーム間で重心点を結んだ線分が計数線を跨いだ場合、対象の人物はバスに乗車したと推定して計上する。最後に、設定したバスの停車範囲からバスが移動した場合、本処理を終了し、バスに乗車した人数とそのバス会社を出力する。

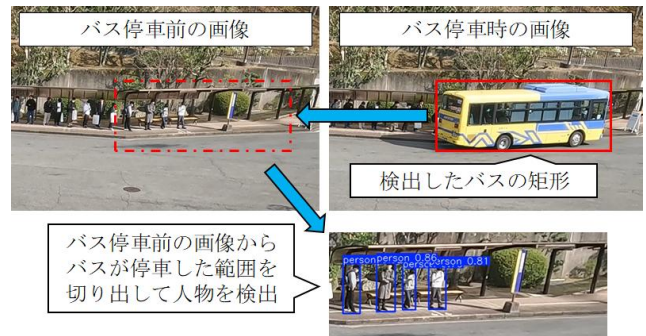


図 11 バスの陰に隠れた人物の計上の流れ

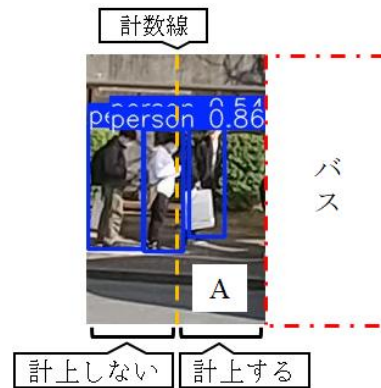


図 12 計数線を越えている人数の補完

3. 実証実験

3.1 実験内容

本実験では、動画像に対して、提案手法を用いてバスの降車人数とバスの乗車人数を計数し、その結果を目視で確認した正解データと比較して、提案手法の有用性を確認する。また、バスの降車人数は、既存手法[4]の計数結果とも比較する。動画像は、画質を 4K (3,840 x 2,160)、フレームレートを 30fps、実際のビデオカメラの設置位置を想定して、図 13 に示すように 2 階から斜めに見下ろし、バス



図 13 撮影の様子

が停車する範囲がビデオカメラの画角に収まるようにロータリーのバス停を晴れの日の日中に撮影した。撮影時間は、バスの降車人数の計数では、バスが到着してから乗客が全員降りきるまで、バスの乗車人数の計数では、バスが停車

してから乗客が乗り込み、再びバスが動き出すまでの間とした。また、動画像はバスの降車人数とバスの乗車人数それぞれ5種類の動画像を検証対象とした。

3.2 結果と考察

バス降車人数を本手法と既存手法で計数した結果を表1に示す。表1を確認すると、バス会社の推定は全て一致し、提案手法の適合率、再現率とF値は全ての動画像で0.920以上の値となった。また、既存手法の計数結果と比較しても、全ての動画像で同等かそれ以上の値であることがわかる。以上のことから、提案手法の有用性を確認できた。さらに、図14に示すように、提案手法ではバスの窓に反射した人物を検出していないため、バスの窓に反射した人物の影響による誤計数の可能性を低減できたと考えられる。しかし、既存手法と同様にバス降車口付近の通行人の影響によって人物の追跡が入れ替わり、誤計数する課題は残存しているため、今後は通行人の影響を受けないバスの降車人数の計数手法を検討する必要がある。

バス乗車人数を計数した結果を表2に示す。表2を確認すると、バス会社の推定は全て一致し、提案手法の適合率、再現率とF値はB4を除いて0.900以上の値となった。このことから、提案手法の有用性を確認できた。しかし、過剰計数や計数漏れも見受けられた。本手法は、バスの車体の後方にある乗客のみを計数するアルゴリズムであったため、通行人を誤って計上し、過剰計数となった。これに対しては、バスの車体前方に通行人を計数するアルゴリズムを組み込むことで解消できると考えられる。計数漏れは図15に示すように、オクルージョンの影響によって乗客を正確に計上できない場合があった。これに対しては、ビデオカメラの設置位置を調整することで、オクルージョンの影響を低減できると考えられる。

本実験で用いた動画像は、晴れの日の日中に撮影したものであるため、夕暮れ時の光量が少ない環境や、雨天天下での雨粒や通行人の傘による提案手法への影響を調査し、提案手法への影響と改良の必要性を調査する必要がある。また、本実験では、ビデオカメラの設置位置が提案手法へ与える影響が明らかとなっていない。そのため、今後は撮影条件を変化させることで、バスの乗降者人数をより正確に計数できるのか調査する必要がある。

4. おわりに

本研究では、バス停を撮影した動画像からバスの降車人数を計数する際に、バスの窓の反射による誤計数を低減する手法とバスの乗車人数を計数する手法を提案した。そして、実証実験により、提案手法の有用性を確認できた。今後は、多様な条件下で撮影した動画像を用いて検証することで、提案手法の適用可能な範囲を明確にしつつ、実用可能性を高めるための改良を続ける予定である。

参考文献

- [1] 国土交通省, “令和7年度版交通施策白書(トピック)”, <https://www.mlit.go.jp/sogoseisaku/transport/sosei_transport_fr_000191.htm>, (参照 2025-05-29).
- [2] 中島 颯人, 新井 イスマイル, 藤川 和利, “バス車載ドライブレコーダを用いた背景差分法による乗降客数推定手法の評価”, マルチメディア, 分散, 協調とモバイル(DICOMO 2018)シンポジウム論文集, Vol.2018, p.43-48 (2018).

表1 バス降車人数の計数結果

動画像	手法	バス会社推定結果	正解人数	計数人数	計数内正解人数	適合率	再現率	F値
A1	提案	一致	51	54	50	0.926	0.980	0.952
A2			40	41	40	0.976	1.000	0.988
A3			49	49	49	1.000	1.000	1.000
A4			22	22	22	1.000	1.000	1.000
A5			12	12	12	1.000	1.000	1.000
A1	既存	一致	51	54	50	0.926	0.980	0.952
A2			40	40	40	0.975	0.975	0.975
A3			49	50	48	0.960	0.980	0.970
A4			22	22	22	1.000	1.000	1.000
A5			12	12	12	1.000	1.000	1.000

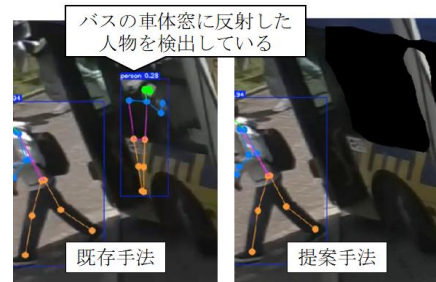


図14 既存手法と提案手法の比較

表2 バス乗車人数の計数結果

動画像	バス会社推定結果	正解人数	計数人数	計数内正解人数	適合率	再現率	F値
B1	一致	38	39	37	0.949	0.974	0.961
B2		40	38	36	0.947	0.900	0.923
B3		37	39	37	0.949	1.000	0.974
B4		30	23	23	1.000	0.767	0.868
B5		14	15	14	0.933	1.000	0.966



図15 オクルージョンの影響による検出ミス

- [3] 山田 遊馬, 廣森 聡仁, 山口 弘純, 東野 輝夫, “測域センサを利用した高精度な路線バス乗降計数システム”, 情報処理学会論文誌, Vol.60, No.3, p.934-944 (2019).
- [4] 中原 匡哉, 松井 友哉, “深層学習を用いたビデオカメラ映像からのバス降車人数の計数手法に関する研究”, AI・データサイエンス論文集, Vol.5, No.3, pp.434-443 (2025).
- [5] Joseph, R., Santosh, D., Ross, G. and Ali, F., “You Only Look Once : Unified, Real-Time Object Detection”, *CVPR*, pp.779-788 (2016).
- [6] Ultralytics, “YOLOv8”, <<https://github.com/ultralytics/yolo-v8>> (accessed 2025-05-29).
- [7] Ultralytics, “YOLO11”, <<https://github.com/ultralytics/ultralytics>> (accessed 2025-05-29).