

路線バスにおけるカメラ画像を用いた OD計測システムの提案

山田 遊馬^{1,a)} 廣森 聡仁^{1,b)} 山口 弘純^{1,c)} 東野 輝夫^{1,d)}

概要: 地域交通の要である路線バスにおいて、乗客トリップ(乗降停留所)情報を把握することは、地域公共交通の利用状況理解と業務改善に大きく貢献する。乗客トリップの把握には交通系ICカード等の情報が有効であるものの、導入コストが高いため赤字路線を多く抱える地域路線バスへの導入は容易ではなく、また定期券や特別バスなどの乗客には対応できない。本研究では、バス車内に設置された単一のカメラ画像のみから各乗客の乗降した停留所を把握する、低コストな乗客トリップ計測システムを提案する。提案手法では、カメラ映像から一定時間間隔で切り出したフレーム毎に乗客の顔認識を行い、前後のフレーム間で検出された顔画像から同一人物を検出しトラッキングを行う。認識の時間間隔を適切に設定することで顔認識にかかる負荷を軽減でき、CNNを活用したコンパクトな深層学習モデルを活用することで計算時間削減を図っている。これらにより、安価なシングルボードコンピュータでもリアルタイム動作可能なエッジ計算型アルゴリズムを実現し、画像情報を外部クラウドサーバに送信する方法におけるプライバシー懸念を解消するとともに通信量も大きく抑制できる。バス会社の協力のもと、実際の路線バスに実験用人員を乗せて走行した車両の車内映像を用いて性能評価を行った結果、最良のモデルでAUC = 0.854、処理時間 0.70s を達成し、十分実用可能な評価結果が得られた。

1. はじめに

地域交通による人の移動情報は、地域社会に対するMaaS(Mobility as a Service)のための基礎データとして近年ますます必要性が高まっている。MaaSでは人々の詳細な交通需要をマイクロレベルで把握し、マルチモーダル交通体系を効率よく構築することが重要視される。また、詳細な移動需要が把握できれば、既存交通システムの経路・運行最適化、沿線におけるイベント開催による不採算経路の利用活性化など、より安全で豊かな地域社会を実現するために有効な施策に寄与する。

中でも、地方都市や郊外・過疎地域など、あらゆる地域において公共交通の軸である路線バスの利用状況把握は、地域公共交通の利用状況の理解と正しい改善プロセス実現のためには極めて重要である。特に、乗客の発着地の計測は、乗客数を正確に把握できていない交通事業者にとって貴重なデータとなる。加えて、例えば混雑路線で多数の乗

客が一度に降車し停留所での停車時間が増大したり、多数の観光客が現金利用や支払い方法で戸惑い、一人あたりの降車時間が増加したりする現象などは、乗降に要する時間を計測し解析することで把握できる。そういったデータはスムーズな運行が実現できているかの指標となり、それに基き事業者が改善を図るための基礎データとなり得る。

主要都市においては、人々の交通機関の利用実態を把握するため、約10年に1度の頻度でパーソントリップ調査(PT調査)が実施されているものの、PT調査は人手に頼っているために人的コストが高く、時間帯や曜日、季節に応じて時々刻々と変化する人々の行動を把握することが困難である。また、地方都市ではPT調査そのものが実施されない現状がある。一方、交通系ICカードの利用履歴を用いることで、発着地(OD; Origin-Destination)を含んだバストリップ調査を行うことも不可能ではない。しかし、特に赤字路線を抱える地域交通においては経営上の理由によりICカードを導入していないことも多く、導入していても一部路線に限られる場合も少なくない。また、ICカードを所持していない乗客の計測は不可能である。

路線バスにおける乗降計測のため、赤外線センサを用いたシステムが開発されている[1]。赤外線センサを2台並べて用い、それらの通過検知時刻の差を利用することで、乗車・降車別の乗降数カウントを実現する。しかし、複数

¹ 大阪大学大学院 情報科学研究科
Graduate School of Information Science and Technology, Osaka University
a) yuma-ynd@ist.osaka-u.ac.jp
b) hiromori@ist.osaka-u.ac.jp
c) h-yamagu@ist.osaka-u.ac.jp
d) higashino@ist.osaka-u.ac.jp

人が同時にセンサを通過する場合は正確に計測できず、また乗降に要する時間も計測できない。我々の研究グループでは、測域センサ (LiDAR; Light Detection and Ranging) の検出データを Raspberry Pi でリアルタイム処理する、軽量かつ高精度な路線バス乗降計測システムを提案している [2]。しかし、本システムを含め、赤外線を利用する方式では OD 計測を目的としていない。

本研究では、路線バスなどの車内を俯瞰するような位置に設置した単一カメラからの画像と Raspberry Pi のような安価なシングルボードコンピュータのみで、各乗客の OD (乗車および降車停留所) をリアルタイムに検出する低コストな OD 計測システムを提案する。提案手法は、各停留所間の運行中 (以下、単に運行中という) に車内の乗客を認識しておき、停留所前後での車内の乗客を比較することで、その停留所で降りた車内の乗客ならびにその停留所で乗った新しい乗客を把握する。これにより、各乗客の OD 計測が可能となる。

カメラ画像を用いる主な課題として、乗客と乗客が重なることでカメラから遠い乗客が視認できなくなるオクルージョン問題があげられる。また、顔の角度や光によっては検出や認識に失敗することもある。そこで、提案手法では、各運行中に一定間隔 (数秒程度) で撮影された複数枚のカメラ画像を用いることで、これらの影響をなるべく抑制する。一方で、複数枚のカメラ画像間に映る同一人物の特定を常時行う必要があるため、2つの人物画像が同一人物か否かを軽量かつ高精度に判別するアルゴリズムを提案する。同アルゴリズムは、深層学習を用いて人物の顔画像から特徴量ベクトルを算出する特徴量抽出モデルを公開データセットで構築する。カメラ画像に映る人物の顔画像を検出し、顔画像の特徴ベクトルを比較することで、同一人物か否かを判定する。

関連研究として、バスの乗降口付近の天井に RGB カメラを設置し、画像処理をすることで乗降者を検出するシステムが開発・商品化されている [3]。そういったシステムでは天井から床面方向にカメラを設置しているため、カメラ視野における乗客同士の重なりによるオクルージョンを回避できる。しかし、これらはいくまで乗客の乗降数計測が目的であり、OD 計測はできないため、提案手法とは異なる。OD 計測を単体カメラでリアルタイムに実現するシステムは我々の知る限りこれまでに提案されていない。

提案手法の有効性を評価するために、実際の路線バスを再現した環境で撮影されたカメラ画像を用いて性能評価を行った。50種類の学習モデルを作成し Raspberry Pi 3 を用いて性能評価を行った結果、最良のモデルで $AUC = 0.854$ 、処理時間 0.70s を達成し、従来の顔画像識別システムと遜色ない精度かつ高速に処理できることが示せた。

2. 関連研究

屋内外の人の通行を把握する取組が多数実施されており、様々なセンサが活用されている [4], [5], [6], [7]。代表的な取組として、赤外線センサを用いた取組、カメラを用いた取組、測域センサを用いた取組が挙げられ、それらについて以下で紹介する。

2.1 赤外線センサを用いた通行者カウント

安価な赤外線センサを用いて、通行者を計測する手法がいくつか提案されている [8], [9]。文献 [8] は、空港内の代表的なボトルネックである保安検査場における乗客数と通過時間の計測を、赤外線センサとマット型の圧力センサを組み合わせた安価なシステムで実現できることを示している。また、文献 [9] では、人が通過する専用のゲートに集電型赤外線センサを設置し、そのゲートを通行する人の数と方向を計測する手法が提案されている。複数人が間隔 30cm の近接縦列で通行する評価実験では、ほぼ 100% の精度で人の通行を検知ができること、講義室への入退出する人を対象とした評価実験では、88% 以上の精度で入退出する人の数を計測できることを示している。また、赤外線センサを用いた通過人数計測システムが製品化されている [1]。この製品では、壁面に二台の赤外線センサを設置し、人が通過した際に生じるセンサ検知の差分から、人が通行する方向を判断している。これらの手法は、小型の赤外線センサを利用しており、様々な場所に簡単に設置できる一方、赤外線センサの検知範囲が限定されているため、ある程度広い領域での計測は適しておらず、また、個々の人の通行を区別することはできないため、バスへの乗降を計測することは困難である。例えば、近年は幅の広い乗降口をもつノンステップバスが広く普及しており、乗客の一人一人が列をなして乗降するような状況だけでなく、複数人が並んで乗降する場合も少なくない。また、整理券の取得や IC カードの接触のために、乗降口付近にしばらく立ち止まる状況も多く、これらの挙動を二台のセンサだけで検知することは困難であり、センサ間の通行検知時刻の差に基づく通行者検知は必ずしも有効な手法ではない。

2.2 カメラを用いた通行者カウント

通行者の計測のために、RGB カメラやステレオカメラを用いた手法も提案されている [10], [11]。これらの手法では、対象領域における人の位置を適切に把握できるように、天井にカメラを設置し、通行する人の数を計測している。文献 [11] では、天井に設置された監視カメラで撮影された映像に基づき、通行者の数を計測する手法が提案されている。この手法は、HOG 記述子に基づき、映像内の歩行者を検出するとともに、カルマンフィルタによって歩行者の

移動方向を検出し、人の入退室を計測する手法で、91%以上の推定精度を達成している。また、RGBカメラを用いた乗降者数カウントシステムも製品化されており、推奨環境下では約95%の推定精度を達成している [3]。同様に、赤外線LEDとステレオカメラを用いたシステムも製品化されている [12]。これらの手法では、高い精度で通行者の数を計測できる一方、各停留所の乗降者数しか計測ができず、ODの計測ができないといった欠点がある。

2.3 測域センサを用いた乗降計測システム

我々の研究グループでは、導入に関する労力やコストの障壁を低減し高精度な検出を実現するため、水平に設置した測域センサを用いた乗降計測システムを開発している [2]。本システムは、背景差分法を用いて物体表面に相当する点群を抽出するとともに、その点群から人体表面に相当する点群を検出する。それに基づき各乗降客の点群を追跡し、扉通過を検知することで乗降者数の計測を実現している。しかし、2.2節と同様に、各停留所の乗降者数しか計測ができず、ODの計測ができない。

2.4 本研究の位置づけ

提案手法は、カメラ画像と軽量な人物識別モデルを用いて、運行中の車内の乗客を把握し、停留所の前後の車内の乗客を比較することで、2.1-2.3節の手法ではできない乗客のOD計測を実現している。また、提案手法は、カメラ画像を安価なエッジ端末でリアルタイム処理し、処理結果(匿名のOD情報)のみをクラウド上にアップロードする。認識した顔画像や特徴量は乗客が降車したと認識されれば即時に廃棄できるため、既存のオフライン動画画像を用いるシステムと比較してもバス車外に画像がでることもなく、プライバシーに配慮した運用が可能である。さらに、通信量の削減、ハードウェアのコスト抑制といった軽量エッジコンピューティングを実現している。

さらに、本稿で提案する人物画像識別システムでは、人物の事前登録が必要がないため、路線バスのOD計測のみならず、ショッピングモールなどの不特定多数の人々が行き交うような場所にも適用可能である。このように本システムの応用範囲の広さも特徴の1つである。

3. システム概要

本研究で提案するシステムの概要を図1に示す。RGBカメラ、シングルボードコンピュータ(以下、簡単のため単にコンピュータとよぶ)、GPSセンサを組み合わせたものを1計測システムとし、計測システムを路線バスの前方天井に後方まで見渡せるような角度で設置する。データ転送量削減のため、個々の計測システムには、RGBカメラとコンピュータが接続されており、コンピュータで画像処理をした後に推定したOD計測情報を記録する。また、

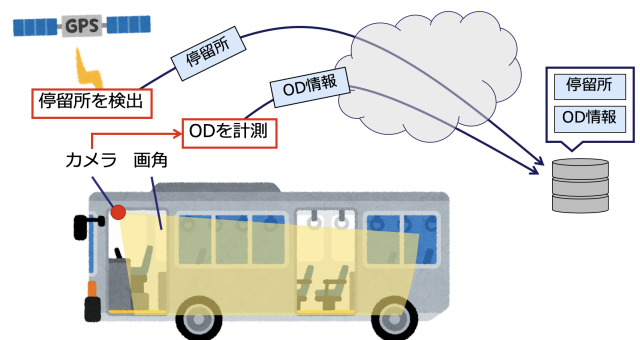


図1: 提案システムの概要

GPSセンサから停留所を検出し、停留所とOD情報を併せてデータサーバにアップロードし、各停留所の乗降情報を集約する。

4. 提案手法

提案手法の概要を図2に示す。まず、バス車両から扉の開閉状態を取得する。閉扉状態であればバスは運行中(停留所間を移動中)とみなし、乗客検出器を動作させる。乗客検出器では、バス車内に設置したRGBカメラから車内画像を取得する。そして、その車内画像に対して顔検出器を適用し、車内の人物(乗客)の顔画像とその画像上の位置をすべて抽出し、抽出した顔画像情報を顔画像メモリ(コンピュータ上のRAM)に追加する。この車内画像の取得から顔画像メモリに追加するまでの一連の動作をバスの運行中フラグがONになっている間繰り返す。そして、バスの扉が開扉状態になり乗降が開始されると、顔画像メモリに格納されている運行中に検出されたすべての顔画像データを人物識別器に入力し、同一人物の顔画像ごとにグルーピングし、その結果を計測データメモリ(RAM)にその時の停留所と併せて記録する。そして、1路線の運行が終了したときに、計測データメモリに格納されている計測データをOD集約器に入力し、それぞれの乗客のOD(乗車した停留所および降車した停留所)を算出し、それをOD情報としてデータサーバに送信する。以降の節では、各機構の詳細について述べる。

4.1 顔検出器における顔画像類似度算出手法

本節では、2つの顔画像から同一人物か否かを判定するために用いる類似度を算出する手法を述べる。本手法で提案する類似度算出モデルを図3に示す。まず、入力された2つの顔画像をそれぞれ特徴抽出モデルに入力し、特徴ベクトルを算出する。そして、算出された2つの特徴ベクトルを比較し、類似度を算出する特徴比較器を適用し、2つの顔画像の特徴量を算出する。4.1.1項に特徴抽出モデル、4.1.2項に特徴比較器の詳細を述べる。

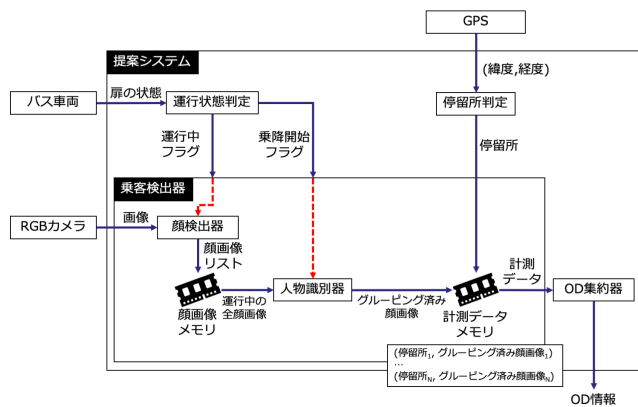


図 2: 提案手法の概要

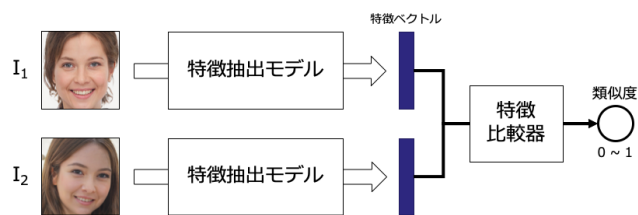


図 3: 顔画像類似度算出手法モデル

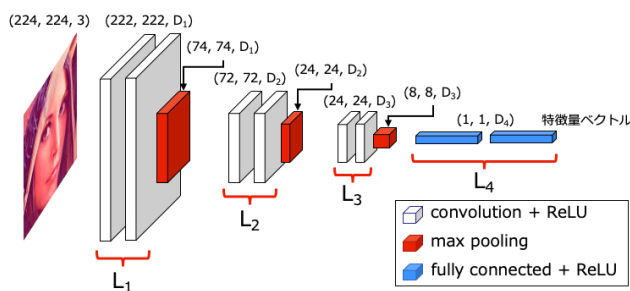


図 4: 特徴抽出モデル

4.1.1 特徴抽出モデル

本項では入力された顔画像の特徴量を抽出するモデルについて述べる。安価なコンピュータで動作させるため、ある程度の精度を保ちつつ、軽量なモデルを構築する必要がある。そのため、本研究ではCNNを用いた深層学習により特徴抽出モデルを実現する。実現したネットワークモデルを図4に示す。白いレイヤーがカーネルサイズ 3×3 の畳み込み層+活性化関数 (ReLU), 赤いレイヤーが Max Pooling 層, 青いレイヤーが全結合層+活性化関数 (ReLU) を表している。特徴抽出モデル自体は畳み込み層と Max Pooling 層を連結させた特徴抽出層を3層連結させ、最下層に全結合層を連結させることで特徴量を表現している。図4の L_i ($i \in 1, 2, 3, 4$) は畳み込み層および全結合層の層数を、 D_i ($i \in 1, 2, 3, 4$) は各層のチャンネル数を表しており、5章の性能評価では、これらのパラメータを変化させ複数のモデルを作成し性能評価を行った。

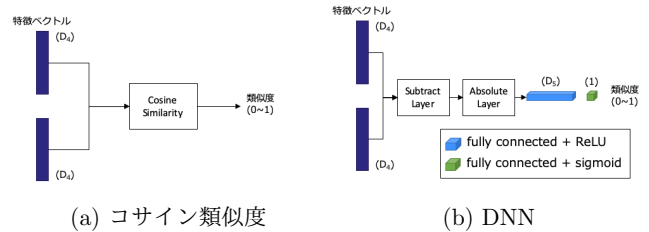


図 5: 特徴比較モデル

4.1.2 特徴比較器

本研究で採用した2つの特徴比較器を図5(a), (b)に示す。図5(a)のモデルは、ベクトル空間モデルの類似性を表すコサイン類似度を適用したモデルである。コサイン類似度は2つのベクトルのなす角の近さを表現しており、しばしば文章などの分類問題に用いられる。図5(b)のモデルは、各特徴の差の絶対値に対して全結合層を2層適用することで類似度を表現するDNNモデルである。この全結合層のパラメータを4.1.1項の特徴抽出モデルと同時に学習させることで、顔画像の比較に特化した比較器の実現が期待できる。

4.2 人物識別器

本節では、運行中に得られた複数枚の車内画像から検出された顔画像を同一人物ごとにグルーピングして出力する手法を述べる。運行中に得られた画像の枚数(総フレーム数)を T , 各フレーム t ($0 \leq t < T$) で検出した人物の数を $N^{(t)}$, それぞれの顔画像を $I_i^{(t)}$ ($0 \leq i < N^{(t)}$) とする。理想的には、運行中に得られたすべてのフレームの顔画像の特徴量の類似度を総当たりで比較することが精度面からは望ましいが、本手法では、計算時間の抑制を重視し、連続する2枚のフレームで検出された顔画像間のみを比較し、類似度を計算する。これをすべての連続フレーム間で行い、最終的に全フレームの顔画像のグルーピングを行う。以降では、連続する2フレーム $t-1, t$ で検出された顔画像を突き合わせする手法について述べる。まず、前後のフレームで検出されたすべての顔画像間の類似度を算出する。運行中のバス車内では、乗客は大きく移動しないため、検出される顔画像の位置はほとんど変化しない。よって、顔画像自体の類似度(4.1節)および後述するフレーム内の位置の類似度を複合させた類似度を算出することで、顔画像類似度算出器の精度が多少荒くとも位置情報を用いることで類似度算出の補正を行い、精度の向上を図っている。顔画像の類似度を $\text{sim}_f(I_a^{(t-1)}, I_b^{(t)})$, フレーム内の位置の類似度を $\text{sim}_l(I_a^{(t-1)}, I_b^{(t)})$ としたとき、顔画像 $I_a^{(t-1)}, I_b^{(t)}$ の複合類似度 $\text{sim}(I_a^{(t-1)}, I_b^{(t)})$ を式(1)のように定義する。ここで、パラメータ α ($0 \leq \alpha \leq 1$) は顔画像類似度の重みを表している。

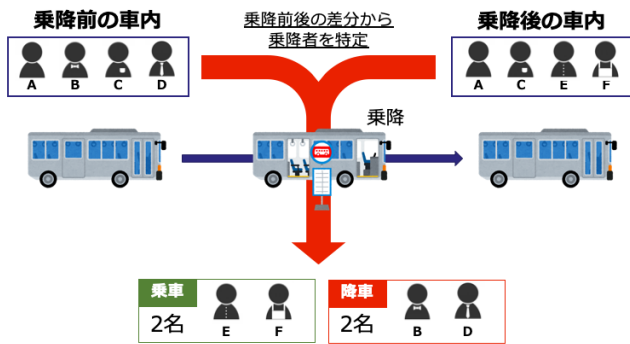


図 6: OD 推定の概要

$$\begin{aligned} & \text{sim}(I_a^{(t-1)}, I_b^{(t)}) \\ &= \alpha \text{sim}_f(I_a^{(t-1)}, I_b^{(t)}) + (1 - \alpha) \text{sim}_l(I_a^{(t-1)}, I_b^{(t)}) \quad (1) \end{aligned}$$

また、フレーム内の顔画像位置の類似度 $\text{sim}_l(I_a^{(t-1)}, I_b^{(t)})$ は、2枚の顔画像間の距離 $\text{dist}(I_a^{(t-1)}, I_b^{(t)})$ を用いて式(2)のように定義する。

$$\text{sim}_l(I_a^{(t-1)}, I_b^{(t)}) = 1 - \frac{\text{dist}(I_a^{(t-1)}, I_b^{(t)})}{\max_{I_i^{(t-1)}, I_j^{(t)} \in I} (\text{dist}(I_i^{(t-1)}, I_j^{(t)}))} \quad (2)$$

そして、式(1)に基づいて、すべての顔画像間の複合類似度を算出し、類似度の総和が最も大きくなるように前後のフレーム間の顔画像の突き合わせを行う。そして、すべての前後のフレーム間において、突き合わせられた画像を集約することで、同一人物の顔画像のグルーピングが完了する。最終的に、グルーピングされたそれぞれの顔画像を出力し、図2中の計測データメモリにグルーピング情報を格納する。

4.3 OD 推定

本節では最終的な乗客の OD を推定する手法について述べる。図6に OD 推定の概要を示す。任意の停留所での乗降前後の車内の乗客の差分を算出することで、当該停留所で乗降した乗客の特定が可能である。

OD 推定の具体的な手法について述べる。乗降前の任意の乗客を p 、乗降後の任意の乗客を q としたときの2者の類似スコア $\text{sim}(p, q)$ を式(3)のように定義する。ここでの乗客とは、4.2節でグルーピングされた乗客を表しており、 $p_{\text{last}}, q_{\text{first}}$ はグルーピングされた顔画像のうち時間的に最後の顔画像と最初の顔画像を表している。また、FaseSim は4.1節の顔画像の類似度を返す関数、 th は同一人物とみなす類似度の閾値を表している。

$$\text{sim}(p, q) = \begin{cases} \text{FaseSim}(p_{\text{last}}, q_{\text{first}}) & (> th) - 1 \\ (\text{otherwise}) & \end{cases} \quad (3)$$

停留所 s での乗降前後の車内の乗客の集合を $P^{(s-1)}, P^{(s)}$



図 7: 実験風景 (被験者は協力者であり一般乗客は含まれない)

とする。そして、式(3)に基づいて、それぞれの乗客間の類似スコアを算出し、 $|P^{(s-1)}|$ 行 $|P^{(s)}|$ 列の類似スコア行列 M を作成する。そして、行列 M の各行各列から高々1セルという制約のもと、セルの総和が最大となるようなセルを選択する。このとき、 i 行のセルが選択されていなければ、乗客 $p_i \in P^{(s-1)}$ が停留所 s で降車したことを意味しており、 j 列のセルが選択されていなければ、乗客 $p_j \in P^{(s)}$ が停留所 s で乗車したことを意味する。また、セル (i, j) が選択されていれば、乗客 $p_i \in P^{(s-1)}$ と $p_j \in P^{(s)}$ が同一人物であることを意味する。なお、この類似度の総和を最大化する問題は割当問題と同等であり、ハンガリアン法(計算量 $O(\max(|P^{(s-1)}|, |P^{(s)}|)^3)$)などを用いて解くことができる。

5. 性能評価

5.1 データセット

5.1.1 訓練データセット

本研究では不特定多数の人々の顔画像比較モデルの提案が目的であるため、訓練データとして、様々な年代・人種が含まれる VGGFace2[13] を採用した。VGGFace2 のデータセットから 100 枚以上の顔画像を持つ人物のうち、500 人をランダムに抽出した。そして、同一人物の顔画像 2 枚を入力とする正例および異なる人物の顔画像 2 枚を入力とする負例をそれぞれ 12,800 組ずつランダムに抽出し、これを訓練データセットとした。また、バス車内は太陽光など外光の有無によって明るさが大きく異なるため、訓練データ画像に画像の明暗を調整するガンマ補正をパラメータ γ を 0.8 - 1.25 の範囲でランダムに適用した。

5.1.2 検証データセット

路線バスにカメラを取り付け、実際の路線バスと同じ状況を再現して実験を行い、実験で得られたカメラ画像を用いて検証データセットを生成した。顔画像類似度算出手法の性能評価(5.3.1 項)については、得られた計 13 人 (735 枚) の顔画像すべての組み合わせ 269,745 組を検証データセットとして性能評価を行った。図7に実験風景を示す。

表 1: 特徴抽出モデルの L パラメータ

モデルタイプ	L_1	L_2	L_3	L_4
A	2	2	2	2
B	1	2	2	2
C	1	1	2	2
D	1	1	1	2
E	1	1	1	3

表 2: 特徴抽出モデルの D パラメータ

チャンネルタイプ	D_1	D_2	D_3	D_4
I	128	128	128	128
II	64	128	128	128
III	64	64	128	128
IV	64	64	64	128
V	64	64	64	64

表 3: Raspberry Pi 3 Model B の性能

項目	仕様
OS	Raspbian 8.0 Jessie
CPU	Quad Core 1.2GHz Broadcom BCM2837 64bit
RAM	1GB
消費電力	7W (最大 12.5W)
質量	45g

5.2 検証モデルおよび評価方法

図 2 中の顔検出器として, YOLOv3[14] を顔検出用にチューニングした YOLO Face[15] を用いた. また, 4.1.1 項で述べた提案手法の特徴抽出モデルのパラメータ L, D を表 1, 2 に示す. これら L, D すべての組み合わせ 25 種に加えて, 過学習を防ぐための dropout 率を 0.10, 0.15 の 2 通りを適用した計 50 種類の特徴抽出モデルを生成した.

また, 4.1.2 項で述べた特徴比較器として, コサイン類似度および DNN の特徴比較モデルを用いた. DNN の特徴比較モデルの全結合層のチャンネル数 D_5 はパラメータ D_4 と等しいチャンネル数とした.

評価方法は, 5.1.2 項で述べたすべての検証データセットに対して類似度を予測し, ROC 曲線の右下面積 (AUC; Area Under the Curve) で各モデルの性能を比較した. ROC 曲線とは縦軸に陽性率 (True Positive Rate), 横軸に偽陽性率 (False Positive Rate) をプロットしたグラフであり, しばしば 2 値分類の有用性判断に用いられる.

また, 検証データの予測には Raspberry Pi Foundation の Raspberry Pi 3 Model B を利用し, モデルごとに類似度予測に要した時間を計測した. 表 3 に Raspberry Pi 3 Model B の性能を示す.

最後に, 4.2 節で提案した人物識別器の性能評価を行う. 乗客数における性能を検証するため, バス車内の乗客が 3, 5, 7, 9 人の場合において, 4.2 節の手法を適用し, グルーピングの正解率を検証した. グルーピングの正解率 acc を

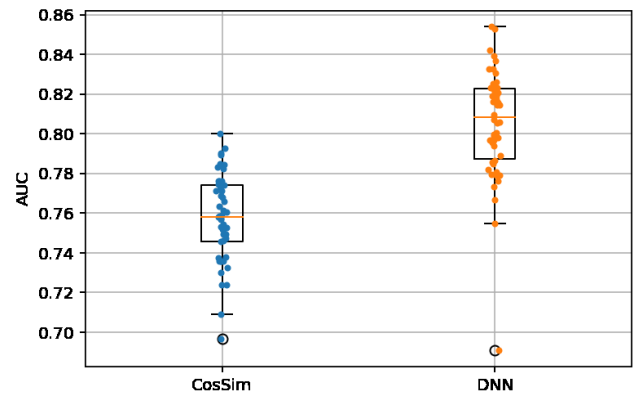


図 8: 特徴比較器ごとの AUC の分布

式 (4) のように定義する. ここで, $G_{tp}^{(i)}$ はグループ i (乗客 i) におけるグルーピングの true positive の数, $G^{(i)}$ はグループ i に分類された画像数を表している.

$$acc = \frac{\sum G_{tp}^{(i)}}{\sum G^{(i)}} \quad (4)$$

5.3 評価結果

5.3.1 顔画像類似度算出手法

はじめに, 4.1.2 項で提案した 2 つの特徴比較器の評価を行う. 図 4 のように 5.2 節で述べた 50 種類の特徴抽出モデルの出力を 2 つの比較器に入力した深層モデルを学習させ AUC を評価した. 図 8 にそれぞれの比較器ごとの AUC の分布を示す. コサイン類似度, DNN を用いたそれぞれのモデルの平均 AUC は 0.758, 0.805 となり, DNN を用いたモデルの精度が圧倒的に良いことがわかる. コサイン類似度の精度が悪い要因として, 特徴ベクトルの次元数が大きくなることで, 次元の呪いが発生し類似度の分離が起りにくくなったことが考えられる. この解決策として特徴ベクトルの次元数を減らすことが考えられるが, 次元数を減らすことで顔画像の様々な特徴が表現しにくくなるため, コサイン類似度はこのモデルに適していないといえる. よって, 以降の性能評価では特徴比較器として DNN を用いることとする.

次に各モデルの性能評価を行う. 図 9 に各モデルパラメータ L ごとの ROC 曲線, 図 10 にパラメータ L ごとの AUC の分布を箱ひげ図を示す. 図 9 中の凡例は, “モデルタイプ-チャンネルタイプ-dropout 率” を示している. この結果より, モデルタイプ D の AUC がどのチャンネルタイプにおいても概ね 0.82 – 0.83 を推移しており安定している. よって, モデルタイプ D が顔画像の特徴抽出に適していると考えられる. また, 最良のモデル (D-I/dropout: 0.10) において, TPR = 0.820, FPR = 0.238 を達成しており, 正例・負例ともに約 80% の精度を達成しており, 十分実運用可能である.

次に, 顔画像の特徴抽出に要した時間と AUC の関係性

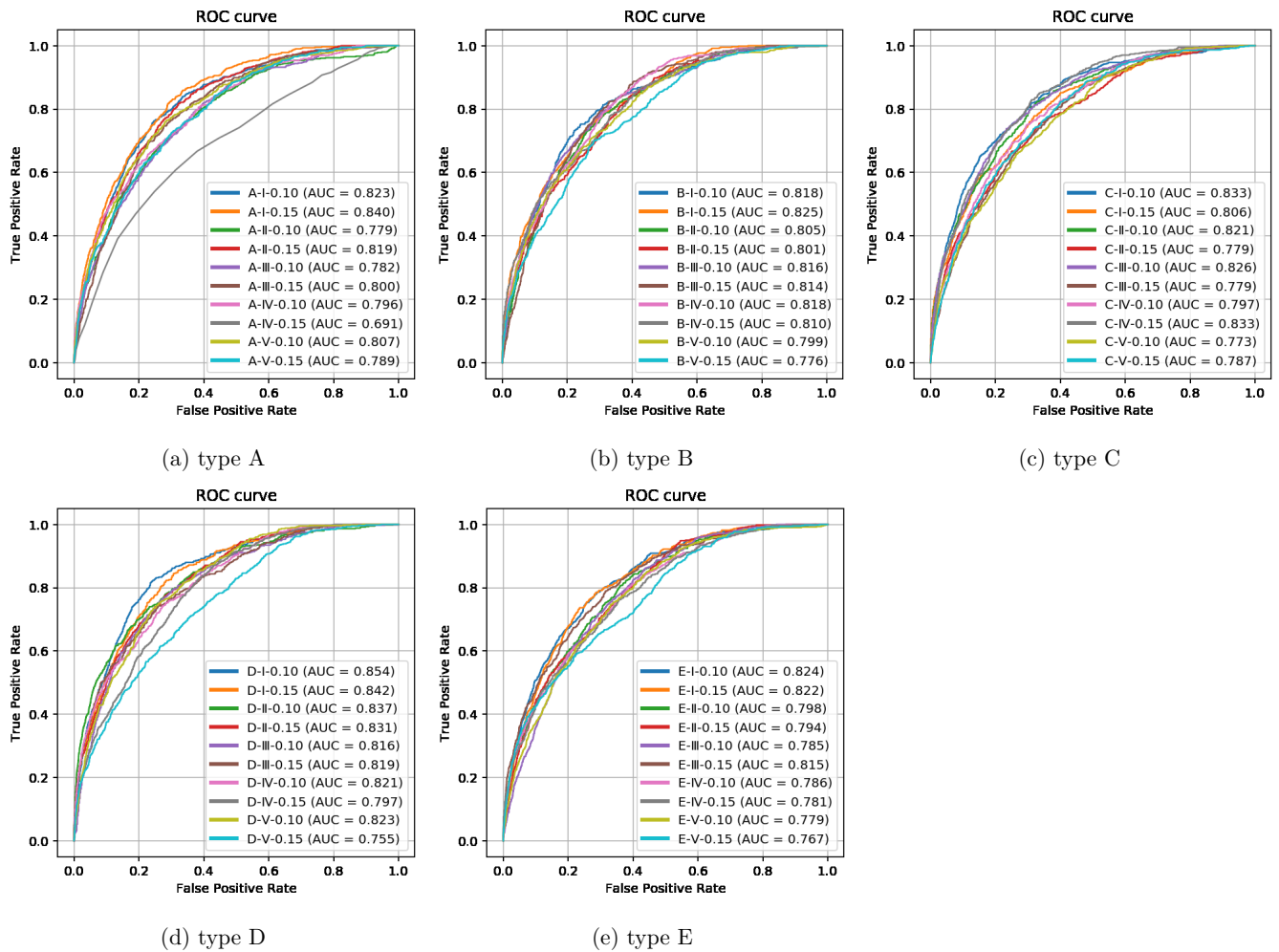


図 9: モデルタイプごとの ROC 曲線

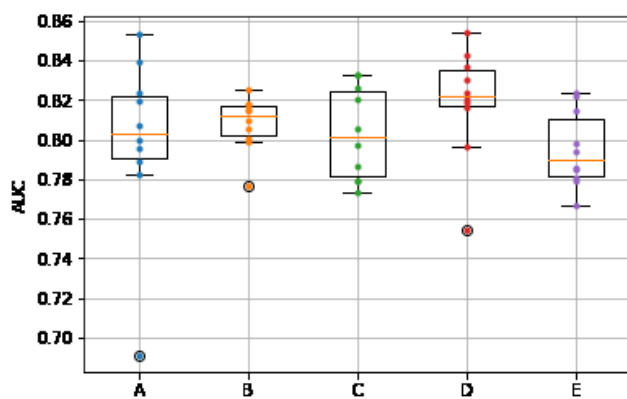


図 10: モデルタイプごとの AUC

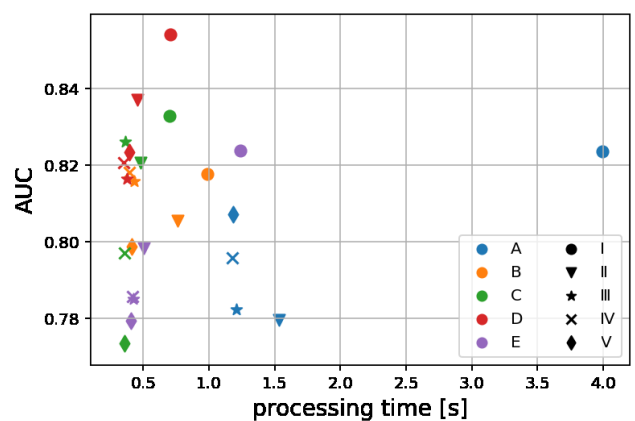


図 11: 処理時間と AUC の関係

について評価をする. 図 11 に顔画像 1 枚あたりの平均処理時間と AUC をモデルごとにプロットしたものを示す. マーカーの色がモデルタイプ, 形がチャンネルタイプを表している. 最も精度が良かった D-I の平均処理時間が 0.70s と概ね良い結果が得られ, 実運用に適していると考えられる. また, 最も CNN の層が多いモデル A-I, A-II については, AUC が良かったものの処理時間が非常に長いため実

運用への適用は難しい.

5.3.2 人物識別器

5.3.1 項で最も精度が良かった D-I を用いて顔画像グルーピングの性能評価を行った. バス車内の乗客が 3, 5, 7, 9 人の場合において, 4.2 節の手法を適用し, グルーピングの正解率で評価を行った. なお, 顔の類似度と位置の類似度の重みパラメータ α は 0.0 - 1.0 の範囲を 0.2 刻みとし

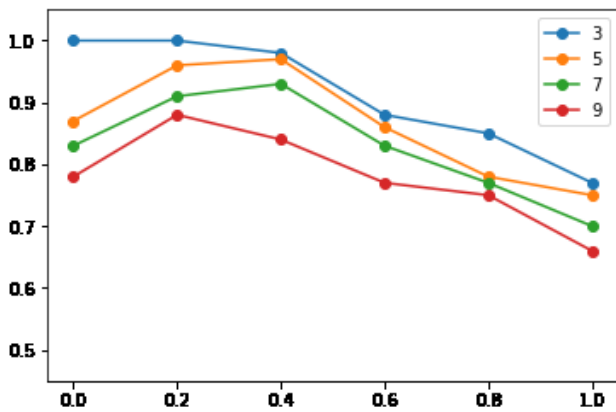


図 12: 乗客数別の顔画像グルーピング精度

た. 図 12 に横軸にパラメータ α , 縦軸に正解率を表した車内の乗客数別の評価結果を示す. パラメータ $\alpha = 1.0$ のときに顔画像比較モデルのみを用いた人物識別結果であるが, 正解率が 60–80% と芳しくなかった. しかし, 位置の類似度の重みを大きくしていくにつれて正解率が向上し, $\alpha = 0.2, 0.4$ のときに約 80–100% の正解率を達成し, 最も精度が良くなっていることが分かる. これより, 顔画像比較モデルにフレーム内の位置情報を付加することで精度向上が図れており, 軽量の顔画像比較モデルを用いた場合でも十分実用可能な精度が得られたと言える.

6. おわりに

本研究では, カメラ画像を用いた路線バスの OD 計測システムを制作するにあたって重要な顔画像の類似度推定手法を提案した. 安価なシングルボードコンピュータでリアルタイム処理ができるように, 深層学習モデルの計算量を減らすため CNN をベースとした顔画像特徴抽出モデルを確立した. 実際の路線バスを再現した環境下で性能評価を行った結果, 最良のモデルで AUC0.854, 1 人あたりの処理時間 0.70s を達成し, 実運用において十分現実的な精度と処理時間を達成した.

本システムでは, 乗客が降車した地点で車載端末上で顔画像を破棄でき, 顔画像をデータとして恒久的メモリに保持することもない. このことから, プライバシー侵害の懸念は少ないものの, 一方で顔画像は個人データ (個人情報) にあたることから, 実運用時には慎重な運用が求められる. 具体的には, トリップ調査目的であること, ならびに個人データは即座に匿名化処理され, 匿名化後のデータを含めて記録はしないことを明示するとともに, オプトアウトの方法もあわせて明示する必要がある. 自治体と連携し, 地域交通改善のための理解を乗客に求めていく必要もあると考える.

今後の課題として, 本稿では顔画像単体を入力とした類似度推定手法を提案したが, 顔画像の他に目や鼻などの顔のパーツ画像を別途入力することで精度の向上かつモデル

の縮小化が期待できる.

参考文献

- [1] Fanbright. 通行カウンター. <http://www.fanbright.jp/service/movingcounter/> (ref. 2017-06-22).
- [2] 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測システム. 情報処理学会論文誌, 第 60 巻, pp. 934–944, 2019.
- [3] 技研トラステム. Passenger counter. http://www.trastem.co.jp/product/passenger_counter.html (ref. 2017-06-22).
- [4] U. Scheunert, H. Cramer, B. Fardi, and G. Wanielik. Multi sensor based tracking of pedestrians: a survey of suitable movement models. In *IEEE Intelligent Vehicles Symposium, 2004*, pp. 774–778, June 2004.
- [5] Ryan Greene-Roesel, Mara Chagas Diogenes, David R Ragland, and Luis Antonio Lindau. Effectiveness of a commercially available automated pedestrian counting device in urban environments: Comparison with manual counts. *Safe Transportation Research & Education Center*, 2008.
- [6] Hong Yang, Kaan Ozbay, and Bekir Bartin. Investigating the performance of automatic counting sensors for pedestrian traffic data collection. In *Proceedings of the 12th World Conference on Transport Research, Lisbon, Portugal*, Vol. 1115, 2010.
- [7] Hong-bo Qian and Hao Han. The applications and methods of pedestrian automated detection. In *Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference on*, Vol. 3, pp. 806–809. IEEE, 2010.
- [8] Dietmar Bauer, Markus Ray, and Stefan Seer. Simple sensors used for measuring service times and counting pedestrians: Strengths and weaknesses. *Transportation Research Record: Journal of the Transportation Research Board*, No. 2214, pp. 77–84, 2011.
- [9] 秦淑彦. 集電型赤外線センサを用いた 1 人用ゲートにおける通行判定. 日本知能情報ファジィ学会誌, pp. 887–898, 2016.
- [10] L. Del Pizzo, P. Foggia, A. Greco, G. Percannella, and M. Vento. A versatile and effective method for counting people on either rgb or depth overhead cameras. In *2015 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pp. 1–6, 2015.
- [11] S. D. Pore and B. F. Momin. Bidirectional people counting system in video surveillance. In *2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pp. 724–727, 2016.
- [12] Eurotech. PCN-1001. <https://www.eurotech.com/en/products/PCN-1001> (ref. 2017-06-22).
- [13] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face and Gesture Recognition*, 2018.
- [14] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, Vol. abs/1804.02767, , 2018.
- [15] Thanh Nguyen. YOLOFace. <https://github.com/sthanhg/yoloface> (ref. 2019-08-20).