Vol.2016-MBL-81 No.24
Vol.2016-ITS-67 No.24
2016/12/9

車載カメラを用いたCNNによる方向別歩行者頭部検出法の提案

原 佑輔[1]　小島 颯平[1]　Moustafa Mahmoud Elhamshary[1]　内山 彰[1]　梅津 高朗[2]　東野 輝夫[1]

概要：本研究では，都市部における人流を歩道ごとに把握する目的で，近年注目を集めている Convolutional Neural Networks (CNN) を用いて，ドライブレコーダー画像から歩道上に存在する歩行者を進行方向別に検出する方式を提案する．歩道上に存在する歩行者の移動方向は車の進行方向に対して前方と後方の 2 種類に大別されるため，提案手法では，遮蔽されにくい歩行者の頭部前方および後方の 2 種類の検出を行う．このため，まず CNN による分類器を構築したうえで，その分類器のニューラルネットワーク構造を利用して，CNN による物体検出を実現する手法の一つである Regions with CNN (R-CNN) を適用した．実際に大阪市内でドライブレコーダーにより画像を収集し，本研究の対象となる頭部検出に合うように性能評価を通じてチューニングを行った結果，前方頭部の Precision が 40.0%，Recall が 58.1%，後方頭部の Precsion が 59.0%，Recall が 43.4%であることが分かった．

Proposal of Head Detection Method Based on CNN Using Drive Recorders

Yusuke Hara[1]　Sohei Kojima[1]　Moustafa Mahmoud Elhamshary[1]　Akira Uchiyama[1]　Takaaki Umedu[2]
Teruo Higashino[1]

1. はじめに

都市計画，安全支援，マーケティングなど，様々な目的において都市部における歩行者の分布および移動状況（人流）を把握することは重要である．例えば，把握した人流から人気のあるスポットを検出したり，混雑状況の監視・予測に基づく人流誘導を行う他，災害時の帰宅困難者の人流に応じた救援計画の立案にも活用できると考えられる．

このような人流や人々の分布状況を把握するため，これまでに様々な手法が提案されている．例えばモバイル空間統計 [1] では携帯電話の通信統計情報を用いて区画毎の人口推定を行っている．また，混雑度マップ [2] では GPS 対応の携帯電話利用者から許諾を得て送信される位置情報の分布からの人口推定を行っている．しかし，いずれも 250m メッシュなど一定範囲ごとの人密度を推定する手法であり，“ある道路の左側を駅方向に歩く人数”といったスポット的な人流を把握する試みは見当たらない．一方，防犯カメラ映像を用いて混雑状況を推定する手法 [3], [4] も存在する．これらの固定カメラを用いた手法で都市部全体の人流を把握するためには，膨大な数のカメラを設置する必要があり，設置場所やコストの制約上，現実的ではない．

そこで我々の研究グループでは，近年普及が進んでいる車載カメラの映像を用いた歩道ごとの人流推定法を検討している．様々な道路を走行している複数の車両で撮影された画像から，歩道の歩行者を検出することで人流を推定し，各車両の位置情報と共にサーバーで集約・統合することで，都市部全体における人流把握の実現を目指す．本研究ではこの目標を達成するための基本要素である，撮影された映像から方向別に歩行者を検出する方式を提案する．歩行者の検出には安全運転支援を目的とした歩行者の全身検出のシステム [5], [6], [7] を使用することが考えられるが，歩道上にいる歩行者は車と歩道の間に存在する植え込みや柵などによって身体の一部が遮蔽される事が多いため，これら

[1] 大阪大学 大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
[2] 滋賀大学 経済学部
Faculty of Economics, Shiga University

Vol.2016-MBL-81 No.24
Vol.2016-ITS-67 No.24
2016/12/9

の手法をそのまま適用することは困難である．一方，提案手法では，遮蔽されにくい頭部を検出対象とすることでこの問題を解決する．また，歩道上でほとんどの歩行者は前方と後方のいずれかにのみ移動するため，歩行者の頭部を前方と後方の 2 種類に分類することで歩行者の移動方向を推定する．

文献 [8] において，我々は Haar-Like 特徴量を用いた歩行者頭部検出法を考案した．しかし，近年の画像認識分野では畳み込みニューラルネットワーク (Convolutional Neural Network: CNN) を物体検出に用いた手法が数多く提案され，Haar-Like 特徴量などの従来手法を上回る精度を達成できることが報告されている [9]．そこで本研究では，歩道ごとの人流推定を行うにあたっての基礎となる，映像からの方向別歩行者数の推定を実現するため，CNN による方式を検討する．このため，CNN による歩行者の頭部分類器を構築し，パラメータやネットワーク構造，学習データ量などの影響を評価し，チューニングを行う．さらに，構築した分類器と同一のニューラルネットワーク構造を用いて Regions with CNN (R-CNN) による方向別の歩行者頭部検出を実現し，様々なパラメータを変化させることによる検出性能の変化について検討を行う．その結果前方頭部の Precision が 40.0%，Recall が 58.1%，後方頭部の Precsion が 59.0%，Recall が 43.4%であることが分かった．

2. 関連研究

2.1 車載カメラを用いた人検出

自動運転車に関連する技術の発展とともに，安全運転支援を目的として，車載カメラを用いた歩行者検出法が数多く研究されている．これらの手法は，人の動きを検出する方式と人の形状を検出する方式の 2 種類に大別される．文献 [5] では人特有の動きのパターンを特徴量として歩行者を検出する．しかし，この手法は動きのパターンを抽出するために歩行者の足が一定時間見えている必要がある．また，人の動きを用いて検出を行っているため静止している歩行者は検出することができない．

一方，人の形状を特徴量として歩行者を検出する手法は移動している人と静止している人の両方を検出することができる．文献 [6] ではウェーブレット解析 [10] と Support Vector Machine(SVM)[11] を用いて歩行者検出を行っている．これらの手法は運転支援を目的としており，群衆中では人同士の重なり (オクルージョン) が大きく影響し，検出精度が低くなるという問題が生じる．

2.2 CNN を用いた物体検出

CNN を用いた物体検出は，ImageNet[12] で注目を集めて以来，様々な方式が考案されている．中でも CNN を用いた画像中に複数存在する可能性のある複数クラスの対象物検出は Localization and Classification と呼ばれ，難しい問題の一つである．R-CNN[13] は CNN を用いた複数クラスの対象物検出手法の一つであり，Selective Search[14] により物体の候補領域を抽出したうえで，CNN による分類を行う．これによって，単純なスライディング・ウィンドウを用いた総当たりでの分類よりも高速に物体検出を行うことができる．本研究では，車載カメラにより取得した画像から方向別の頭部検出を行うため，R-CNN を適用する．CNN や R-CNN では適用対象によってチューニングが必要となるため，本稿ではニューラルネットワークの構造やパラメータ設定について評価を通した検討を行う．

図 1 人流推定法の概要

3. 人流推定法の概要

現在検討している人流推定法の概要を図 1 に示す．車載カメラの映像は，一部の協力ユーザから提供されるものとする．車載カメラとしては，ダッシュボードにマウントされたスマートフォンや一般のドライブレコーダーを想定している．ドライブレコーダーの中にはスマートフォンや車載器などと WiFi により接続できる製品が存在する．したがって，携帯通信網により外部ネットワークに接続されたスマートフォンや車載器をゲートウェイとすることで，ドライブレコーダーの映像をサーバーに送信できる．ただし，通信量をできるだけ抑えることが望ましいため，本研究ではドライブレコーダーで取得した映像に対して，スマートフォンや車載器で処理を行い，方向別の歩行者数を推定した後，その結果のみをサーバーに送信することを想定している．方向別歩行者数の推定結果は歩道の位置と撮影時刻とともにサーバーに送信される．サーバーでは，複数車両から送られてきた各地の歩道における人流を統合し，地図上にマッピングする．

Vol.2016-MBL-81 No.24
Vol.2016-ITS-67 No.24
2016/12/9

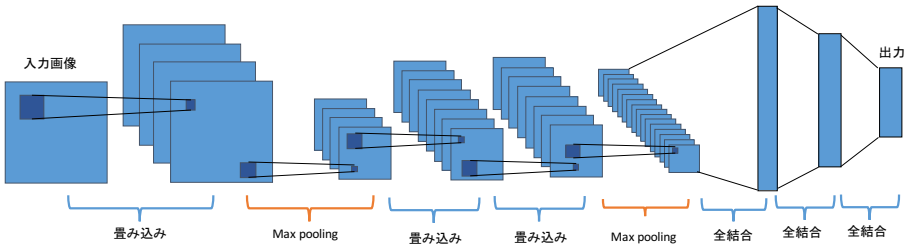


図 2　CNN による典型的な分類器のネットワーク構造

歩道の位置は GPS などにより得られた車両位置と映像内の歩行者の検出位置（左側または右側）から容易に把握可能である．ただし提案手法では頭部のみを検出対象とすることから，映像内の歩行者が小さいと検出に十分な特徴量が得られない．このため本研究では，走行車線に近い側の歩道（日本では左側）でかつ車両から一定距離内のみを推定対象とし，車両が歩道に最も近い車線を走行している時のみ，人流の推定を行う．

本研究では車載カメラにより撮影された映像の 1 フレーム（静止画）に対して，方向別の歩行者頭部がどこに存在しているかを推定することを目標とする．これは画像内の物体の場所とクラス（どこに何があるか）を決定することに等しく，画像処理の分野では Localization and Classification と呼ばれる問題である．これを実現するためには，まず良い Classifier（分類器）を構築し，同一のニューラルネットワーク構造を用いて Localization and Classification を行うアプローチが取られることが多い．このため，本研究ではまず CNN による頭部分類器を構築し，様々なパラメータやニューラルネットワーク構造などを変化させながら Classifier の評価検討を行う．その後，CNN を用いた Localization and Classification の一方式である R-CNN による方向別の歩行者頭部検出を行う．なお，本研究では，歩道上の歩行者は前方・後方のいずれかのみに移動するものとし，検出対象となる頭の方向を前方・後方の 2 種類に限定する．実際には交差点などにおいて道路側を向いて立ち止まっている人なども存在するが，これらの判別は困難なため，本研究では交差点付近を検出の対象外としている．

4. CNN による歩行者頭部分類

4.1 ニューラルネットワークの構造

文献 [15] より，CNN を用いた分類器の構築においては，典型的な例として以下に示すようなネットワーク構造が用いられる（図 2）．

$$[(ConvNet : ActFunc) \times N : Pooling] \times M : \\ (FC : ActFunc) \times K : ActFunc$$

ここで，$ConvNet, ActFunc, Pooling, FC$ はそれぞれ畳み込み層，活性化関数，プーリング層，全結合層を表す．また，N, M, K はそれぞれ畳み込み層，プーリング層，全結合層の数を表す．$ActFunc$ は活性化関数であり，SoftMax や ReLU など様々な関数が用いられる．通常 $N \leq 5$，$0 \leq K \leq 2$ としたネットワーク構造が用いられることが多く，M は様々な値に設定される．一般的に層の数を増やすとネットワークの表現力が上がり，より複雑な分類ができると言われているが，一方で分類器が学習データに特化し過ぎて，それ以外のデータに対して正しい分類ができなくなる，すなわち過学習の可能性が高くなるため，適用対象ごとにチューニングをする必要がある．これらのネットワーク構造を変化させた場合の性能の変化については，6 章で議論する．

4.2 過学習対策

過学習を防ぐため，正規化および Dropout[16] を適用する方法が考案されている．正規化では，コスト関数に $\lambda||w||^p$ で表される正規化項を加えるものである．$||w||$ は各パーセプトロン内の重みの大きさを表したもので，重みを p 乗したものをコスト関数に加えることで重みがなるべく小さくなるように学習され，過学習が起こりにくくなる．特に $p = 1$ の場合を L1 正規化，$p = 2$ の場合を L2 正規化と呼ぶ．一方 Dropout では，それぞれの層内のパーセプトロンを一定割合で無効にする．Dropout を行うことで複数のネットワークを別々に学習することとなる．推定時には全ネットワークを用いるので，複数のネットワークの結果を平均することと同じ効果をもたらし過学習を防ぐと考えられている．本研究でも正規化および Dropout を適用し，過学習をできる限り抑える．

4.3 学習データの拡張

より汎用的な分類をするためには多様かつ十分な数のデータが必要となる．しかし，実際には収集できる学習データの数が限られることが多い．その場合に学習データの拡張を行うことが機械学習ではよく用いられる．画像認識の場合は画像の輝度を変化させたり，回転，平行移動，拡大縮小，反転等をさせて学習データの拡張を行う．

本研究では，頭の向きで移動方向を判別するため，画像を反転させて学習データを拡張することはできない．このため，各学習データの縦横の平行移動，拡大縮小，回転をランダムに行い，学習データを拡張した．平行移動は縦横の長さのそれぞれ 10%の幅で行うものとし，拡大縮小も 10%の範囲で行った．また，回転は ±20° の範囲で行った．

5. R-CNN を用いた方向別の頭部検出

画像中に複数存在する可能性のある頭部の位置と方向を検出するためには，4 章で述べた頭部分類器だけでは不十分であり，画像の中から頭部の候補となる物体の領域を抽出したうえで，それが前方の頭部か，後方の頭部か，それ以外かを分類する必要がある．

本研究では R-CNN[13] と呼ばれる物体検出アルゴリズムを用いる．R-CNN は画像中の物体検出アルゴリズムの一つで，Selective Search[14] を用いて物体が存在する可能性のある候補領域を検出し，それぞれの候補領域を CNN に入力して物体の特徴量を求め，SVM により分類する．本研究では簡単のため SVM を用いず，ニューラルネットワークの出力層を全結合で接続し，SoftMax 関数を用いて出力を行うこととした．R-CNN による方向別頭部検出の概要を図 3 に示す．こうして得られる背景（その他），前方頭部，後方頭部の出力値のうち，最も大きいものを分類結果とする．

5.1 Selective Search 後の追加学習

車載カメラで撮影した画像から Selective Seach で物体候補を探し，学習した CNN で背景，前方頭部，後方頭部に分類を行うと，背景が頭部と分類される誤検出が多くなる．これは学習の際に用意した背景のデータセットが，人のいない風景画像からランダムに切り出したものであるためである．したがって，車載カメラ画像から切り取られた背景や，頭部以外の人身体の画像を背景画像として学習させる必要がある．そこで以下の手順で追加学習を行った．

まずラベル付けされた車載カメラの画像から Selective Search で物体候補領域を抽出する．次に 4 節で構築した CNN 分類器を用いて，抽出した候補領域を背景，前方頭部，後方頭部に分類する．そのうち誤検出であるものを追加で学習を行う．ここでは，真値である頭部領域の 30%以下と重複するような候補領域であるにも関わらず，頭部領域に分類されたものと，真値である頭部領域の TH%以上と重複する候補領域であるにも関わらず，誤ったクラスに分類されたものを誤検出として定義した．

追加の学習方法としては，誤検出されたものを新たなデータセットとして学習率を通常の 0.001(4 節での学習率の 1/10) にして行った．学習率を下げることによって，既に学習されたモデルのパラメータを大きく変えないようにしつつ，誤検出されたデータを考慮してモデルを微調整することができる．

表 1　CNN の学習に用いたパラメータ設定

学習回数	300 [epoch]
最適化アルゴリズム	AdaDelta
学習率	0.01 (100 epoch ごとに 1/10)

6. 性能評価

6.1 実験環境

提案手法の性能評価を行うため，大阪市茶屋町周辺の道路をドライブレコーダ (ユピテル社製 DRY-WiFiV5c) を設置した自動車で複数回通行し，映像を撮影した．撮影は，図 4 の地点 1 から地点 2 間の水色で示されている道路で休日の正午頃に行った．ドライブレコーダーの映像から 30 フレーム毎に画像を切り出し，そのうち 2418 枚を学習データとした．目視で各フレームにおける人の頭部を前方と後方に分類してクリップし学習データとした．学習データの内訳は前方 2100 枚，後方 3500 枚であった．また，BingAPI を用いて道路や風景など人の写っていない写真を 10,574 枚収集し，ランダムなサイズおよび位置でクリップすることで背景用の学習データとした．

分類器のテストデータとして，学習データとは別の画像から人の頭部を前方，後方に分類して合計 1295 枚クリップし，その中から各 600 枚をランダムに抽出して用いた．背景のテストデータもランダムに 600 枚をクリップして用いた．なお，入力画像のサイズは全て 64pixel×64pixel にリサイズして与えた．方向別頭部検出のテストデータは，データとして用いていない車載カメラ画像 734 枚のうち，交差点付近や歩道に最も近い車線を走行していない場合の画像を目視で確認し，除外した後，100 枚をランダムに抽出して用いた．

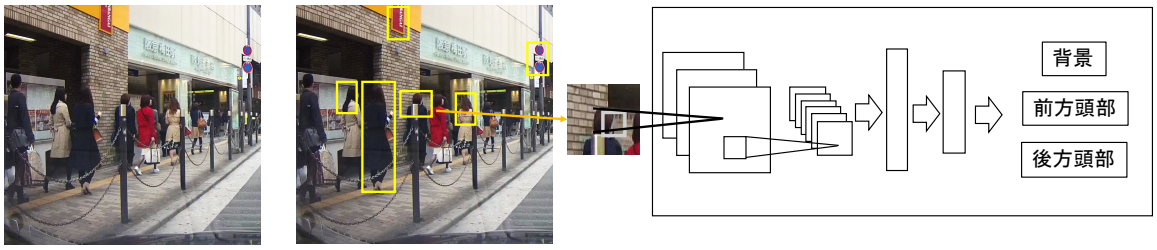
CNN の入力画像サイズは 64×64 とし，入力画像はこのサイズにリサイズしてから，学習や分類を行う．CNN の学習を行う際のパラメータは特に明記しない限り表 1 の設定を用いた．深層学習用ライブラリは，6.2.1 節では Keras(1.0.5) とバックエンドに Theano(0.9.0.dev1) を用いた．それ以外では Nvidia 社製の深層学習用ツール DIGITS(4.0.0) とバックエンドに Caffe(0.15.14) を用いた．学習に用いたワークステーションのスペックは CPU が Intel(R) Xeon(R) CPU E5-1680 v3 @ 3.20GHz，メモリ 128GB，GPU は GeForce GTX 1080 である．

6.2 分類器の性能評価

6.2.1 ネットワーク構造の影響

ニューラルネットワークの構造を変化させた場合の分類精度を比較する．畳み込み層が 2 層でフィルタ数がそれぞれ 32，32 のものをモデル A，畳み込み層が 2 層でフィルタ数がそれぞれ 32，64 のものをモデル B, 畳み込み層が 4 層

Vol.2016-MBL-81 No.24
Vol.2016-ITS-67 No.24
2016/12/9



1. 入力画像　2. Selective Searchによる物体候補領域抽出　3. CNNによる分類

図 3　R-CNN による方向別頭部検出の概要

図 4　実験で走行した道路 (大阪市茶屋町周辺)

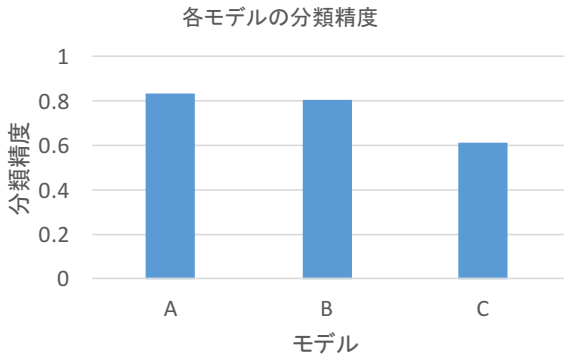


図 5　各モデルの分類精度

でフィルタ数が 4 層とも 32 のものをモデル C とする．各フィルタの大きさは 3×3 とし，活性化関数には LealyReLU を用いた．また，各畳込み層の後には Max pooling 層を置くものとする．Max pooling のサイズは 2 × 2 である．畳み込みの部分の後に全結合層を 1 層おきノード数は 512 とする．活性化関数は LeakyReLU である．最後に出力ノードを 3 つ置き，活性化関数は SoftMax を用いる．

結果を図 5 に示す．モデル C の分類精度が低いが，これは Max pooling を 4 回行ったことで過度なサイズの縮小が行われ，特徴が失われてしまったためと考えられる．実際，モデル A，B，C の総パラメータ数はそれぞれ 3,698,595, 7,394,243，96,227 となり，モデル C は総パラメータ数が小さいために分類性能が低いと考えられる．しかし，A よりも B の方が総パラメータ数が大きいが，分類精度は A のほうが高いために，一概に総パラメータ数が大きければいいとは言えない．そこで以降は ILSVRC2012 で性能トップであった AlexNet[9] を用いることとする．

6.2.2　拡張による学習データ数増加の影響

次に，データ拡張により学習データ数を増加させた場合の影響を調べるため，学習データ数を各カテゴリ 1000 枚から 32,000 枚まで変化させた場合の分類精度を評価した．その際，学習データ全体に対してデータ拡張を適用し，各カテゴリのデータ数が指定した数になるようにした．背景データは十分なデータが存在するため，データ拡張を行わずに必要な数だけをランダムに抽出した．

評価結果を図 6 に示す．拡張によるデータ数の増加に伴い，精度が対数関数的に増加していることが分かる．これは，データ数が増加するほど様々なバリエーションのデータが学習され，より汎用性の高い分類ができるようになるからだと考えられる．

6.2.3　学習データ数の影響

学習データ数の影響を調べるため，拡張前のデータ数を変更し，分類の分類精度を評価した．このため，まず元のデータセット (前方頭部:2100，後方頭部:3500) に対してそれぞれのカテゴリのデータ数を 1/2 にしたデータセット (前方頭部:1050，後方頭部:1750) と 1/4 にしたデータセット (前方頭部:525，後方頭部:875) を作成した．その後，各データセットについて各カテゴリのデータ数が 32000 枚になるようデータ拡張を行い，CNN の学習を行った．

評価結果を図 7 に示す．データ数を 1/2 にした場合は Precision，Recall ともに元のデータセットと比べて若干低くなっている．さらにデータ数を 1/4 にした場合は大幅に下がっていることがわかる．この結果および，6.2.2 節の結

Vol.2016-MBL-81 No.24
Vol.2016-ITS-67 No.24
2016/12/9

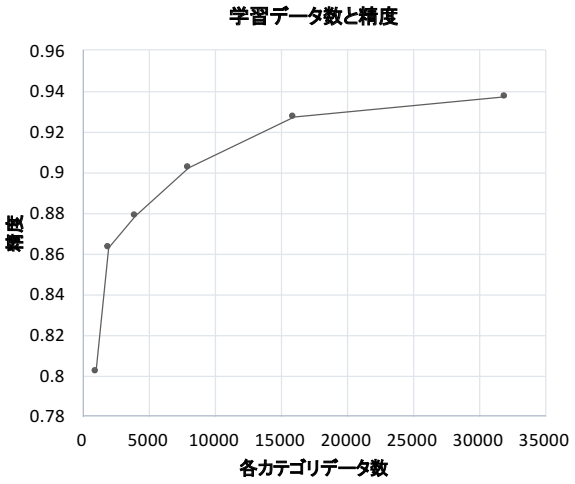


図 6　学習データ数と分類精度

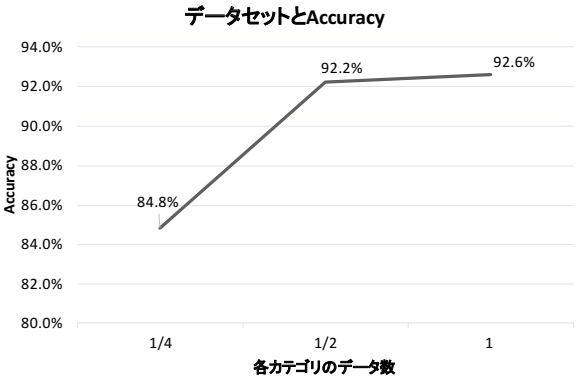


図 7　データ数と Accuracy

果より，データ拡張はあくまで補助的なものであり，実際に撮影したデータを増やすことが精度の向上に大きく貢献していると考えられる．今回用いた学習データは拡張前で前方 2100 枚，後方 3500 枚であるため，拡張前の学習データ数を増やすことにより，さらに精度が向上する可能性がある．現在，我々はデータ収集を進めており，今後拡張前の学習データ数の影響についても評価を進める計画である．

6.3　方向別頭部検出の性能評価

6.3.1　追加学習における学習率の調整

適切な学習率を定めることは，過学習を防ぐために重要である．CNN の学習には表 1 に示す学習率を用いているが，R-CNN による方向別頭部検出においては，追加学習による精度向上を図っている．5.1 節で述べたように追加学習を行う際の学習率は CNN の学習時より小さくする必要がある．この時の適切な学習率を定めるため，追加学習時の学習率の初期値を CNN の学習時と同じ 0.01 にした場合，および 1/10 の 0.001 にした場合で誤差関数の値を比較した．

図 8 は学習率 0.01 の学習過程を示している．なお追加

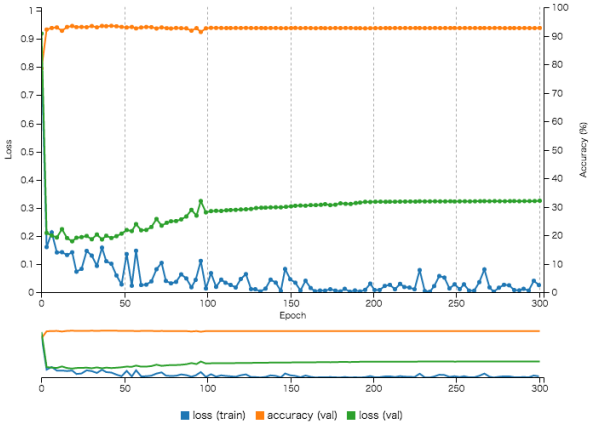


図 8　追加学習時の学習率 0.01 の場合

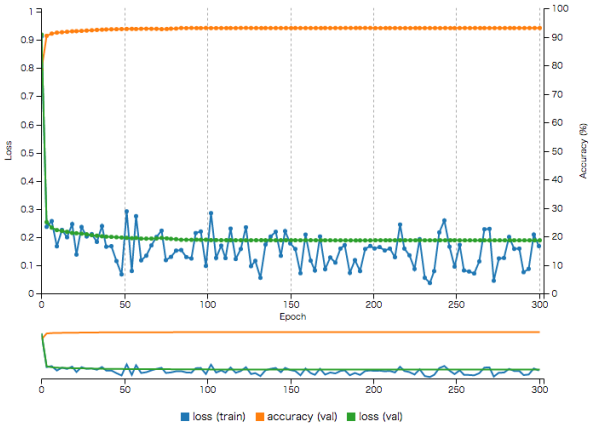


図 9　追加学習時の学習率 0.001 の場合

学習用のデータセットは誤検出された背景と頭部の画像のうち 8 割であり，評価用データセットは残りの 2 割を用いた．青色の学習データセットに対する誤差 (Loss) は低下している一方で，緑色の評価データセットに対する Loss は上昇している．これは既に学習されたモデルのパラメータが大きく変わることで，分類機の作成で獲得した様々な頭部の分類機能が失われ，過学習が起きていると考える事ができる．

一方，学習率を 0.001 にした場合の学習過程を図 9 に示す．青色の学習データセットに対する Loss とともに緑色の評価データセットに対する Loss も低下していることから，過学習を起こしていない．したがって，既に学習されたパラメータを大きく変更せず，Selective Search により抽出される候補領域の特性を考慮した画像に適応できていると考えられる．

6.3.2　追加学習時の正解データ閾値の変化

R-CNN では作成した分類器を頭部検出に適用するために追加学習を行うが，5.1 節で述べたように追加学習用データは Selective Search で抽出した物体候補領域を 6.2 節で作成した分類器で分類した結果のうち，誤検出された背景と頭部の画像である．この誤検出の定義は閾値 TH により調整可能なため，TH を 50%，70%，80%，90%と変化さ

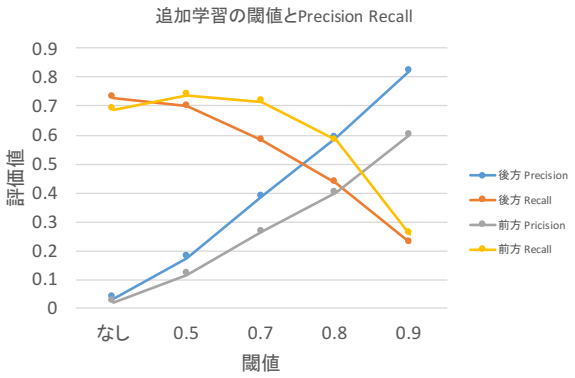


図 **10** 閾値を変化させた場合の Precision Recall

せて評価を行った．テストデータは前述の 100 枚の車載カメラで撮影した画像である．

結果を図 10 に示す．前方頭部，後方頭部ともに閾値を大きくすると Precision が上がり，Recall が下がることがわかる．これは閾値が小さい場合頭部のごく一部の画像を頭部として学習するために，頭部の特徴を捉えきれず誤検出が多くなり，閾値が大きい場合は頭部全体を捉えられている画像のみ学習するために正確に頭部を検出できるが，判定により多くの特徴量が必要となり，検出漏れが多くなるためだと考えられる．全体として，閾値 0.8 のときが比較的バランスが取れており，前方頭部の Precision が 40.0%，Recall が 58.1%，後方頭部の Precsion が 59.0%，Recall が 43.4%であることが分かった．今後データ量を増やすことで性能が向上する余地はあるため，実データの収集をすすめるとともに今回の評価を通じて明らかになった R-CNN による頭部検出の特性を考慮した人流推定法の設計に取り組む．

7. おわりに

本研究では，街中を走行する車両の車載カメラ映像を利用した歩道レベルでの人流推定法を提案した．提案手法では遮蔽されにくい頭部を対象とし，歩行者の頭部前方および後方の 2 種類の検出を行う．このため，まず CNN による分類器を構築したうえで，その分類器のニューラルネットワーク構造を利用して，CNN による物体検出を実現する手法の一つである R-CNN を適用した．実際に大阪市内でドライブレコーダーにより画像を収集し，本研究の対象となる頭部検出に合うように性能評価を通じてチューニングを行った結果，前方頭部の Precision が 40.0%，Recall が 58.1%，後方頭部の Precsion が 59.0%，Recall が 43.4%であることが分かった．

今後，実データ量を増やすとともに，歩道単位での人流推定に対する性能を評価するための実験を計画している．

参考文献

[1] 寺田雅之，永田智大，小林基成：モバイル空間統計における人口推計技術 (社会・産業の発展を支える 「モバイル空間統計」：モバイルネットワークの統計情報に基づく人口推計技術とその活用)，NTT DoCoMo テクニカル・ジャーナル， Vol. 20, No. 3, pp. 11–16 (2012).

[2] 株式会社ゼンリンデータコム：混雑度マップ，`http://lab.its-mo.com/densitymap/`.

[3] Silveira Jacques Junior, J., Musse, S. and Jung, C.: Crowd Analysis Using Computer Vision Techniques, *IEEE Signal Processing Magazine*, Vol. 27, No. 5, pp. 66 – 77 (2010).

[4] Wu, Z., Thangali, A., Sclaroff, S. and Betke, M.: Coupling detection and data association for multiple object tracking, *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1948–1955 (2012).

[5] Wöhler, C., Anlauf, J. K., Pörtner, T. and Franke, U.: A Time Delay Neural Network Algorithm for Real-Time Pedestrian Recognition, *Proceedings of International Conference on intelligent vehicle*, pp. 247–251 (1998).

[6] Papageorgiou, C., Evgeniou, T. and Poggio, T.: A Trainable Pedestrian Detection System, *Proceedings of Intelligent Vehicles*, pp. 241–246 (1998).

[7] Lee, K.-H., Hwang, J. N., Okapal, G. and Pitton, J.: Driving recorder based on-road pedestrian tracking using visual SLAM and Constrained Multiple-Kernel, *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 2629–2635 (2014).

[8] 原 佑輔，小島颯平，内山 彰，梅津高朗，山口弘純，東野輝夫：ドライブレコーダー映像を用いた頭部検出に基づく人流推定法の提案，マルチメディア，分散，協調とモバイル (DICOMO2016) シンポジウム論文集，pp. 253–261 (2016).

[9] Krizhevsky, A., Sutskever, I. and Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems 25* (Pereira, F., Burges, C. J. C., Bottou, L. and Weinberger, K. Q., eds.), Curran Associates, Inc., pp. 1097–1105 (online), available from ⟨http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf⟩ (2012).

[10] 山口昌哉：ウェーブレット解析，科学，Vol. 60, pp. 398–405（オンライン），入手先 ⟨http://ci.nii.ac.jp/naid/10006233574/⟩ (1990).

[11] Bradski, G. and Kaehler, A.: 詳解 OpenCV: コンピュータビジョンライブラリを使った画像処理・認識，O'Reilly Media, Inc. (2009).

[12] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. and Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database, *CVPR09* (2009).

[13] Girshick, R., Donahue, J., Darrell, T. and Malik, J.: Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587 (online), DOI: 10.1109/CVPR.2014.81 (2014).

[14] Uijlings, J. R., Sande, K. E., Gevers, T. and Smeulders, A. W.: Selective Search for Object Recognition, *Int. J. Comput. Vision*, Vol. 104, No. 2, pp. 154–171 (online), DOI: 10.1007/s11263-013-0620-5 (2013).

[15] Li, F.-F., Karpathy, A. and Johnson, J.: CS231n Convolutional Neural Networks for Visual Recognition, `http://cs231n.github.io/convolutional-networks/`.

[16] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.: Dropout:

A Simple Way to Prevent Neural Networks from Overfitting, *J. Mach. Learn. Res.*, Vol. 15, No. 1, pp. 1929–1958 (online), available from ⟨http://dl.acm.org/citation.cfm?id=2627435.2670313⟩ (2014).