

# 入退場者カウント画像処理システムの ニューラルネットワークを用いた精度向上に関する研究

大坪 敦<sup>1,a)</sup> 良永 早耶佳<sup>2</sup> 廣重 法道<sup>1</sup> 鶴田 直之<sup>1,b)</sup>

**概要：**近年、人の移動に関する情報を収集し、マーケティングや防災に活用しようという取り組みが活発になっている。このような背景の中、通行人数の収集を可能とする安価なシステムのニーズが高まっている。そこで筆者らはラズベリーパイとカメラを用いたカウントシステムの開発を行い、福岡県の平成筑豊鉄道や福岡市科学館などで実証実験を行っている。この際のカウント精度 (F 値) は 7~8 割程であった。しかし平成筑豊鉄道側より要求されている精度は 8 割以上であり、これを十分に満たしていない。そこでニューラルネットワークを用いて通過する人物の特徴を学習・認識させる機能を追加することによって、90.5%という精度を達成することができた。

## A Research on improvement counting accuracy for people counter by using neural network

ATSUSHI OTSUBO<sup>1,a)</sup> SAYAKA YOSHINAGA<sup>2</sup> NORIMICHI HIROSHIGE<sup>1</sup> NAOYUKI TSURUTA<sup>1,b)</sup>

**Abstract:** Data of people movements is useful for marketing and disaster prevention. Therefore, recently, requirement for reasonable people counter is increasing. For this requirement, we developed a people count system by using a Raspberry Pi and a USB camera, and applied it to Heisei Chikuhou Railway and Fukuoka City Science Museum. Though an accuracy required from Heisei Chikuho Railway is over 80%, performance of the counting system was from 70% to 80%. In this paper, to achieve the requirement this problem, we propose to add a neural network to the counting system. As a result of this improvement, we achieved 90.5% accuracy.

### 1. はじめに

現代において、人は様々な場所を行き来し生活している。このような人の移動に関する情報はマーケティングや防災に活用できる情報として、近年注目を集めている。こうした背景の中、安価に人の行き来を計測したいというニーズが高まっている。そこでラズベリーパイ [1] と USB カメラを用いた安価なカウントシステム (図 1)(a) の開発を平成 28 年より行ない、福岡県の平成筑豊鉄道や、福岡市科学館で開催される展示会において実証実験を行っている (図 1)(b)(c)[2]。

### 2. 研究目的

カウントシステムの実用化において最も重要視されている点はカウント精度をいかに高めることができるかということである。実証実験を行っている平成筑豊鉄道より要求されている精度は 8 割以上である。これに対し現時点で開発しているカウンタの精度は 7~8 割ほどであり、さらなる精度の向上が求められている。そこで本稿ではニューラルネットワークを用いた精度の向上方式を提案し、その性能評価について述べる。

<sup>1</sup> 福岡大学 工学部 電子情報工学科

<sup>2</sup> 福岡大学大学院 工学研究科 電子情報工学専攻  
Fukuoka University, Jonan-ku, Fukuoka 814-0180, Japan

a) t1141241@cis.fukuoka-u.ac.jp

b) tsuruta@fukuoka-u.ac.jp



(a) カウンタ外観



(b) 平成筑豊鉄道への設置



(c) 福岡市科学館イベント会場への設置

図 1 開発しているカウンタ

### 3. 先行研究

#### 3.1 入退場者カウントシステム

カウントシステムは、ハードウェアに Raspberry Pi3 と USB カメラを使用する。天井にカメラを設置し真下の動画を撮影し、フレーム間差分を用いて人物領域を抽出、分析することによりカウントする。

#### 3.2 フレーム間差分方式によるカウント処理

本方式は前後のフレームの差分を取ることで、移動している人の部分だけを抽出し、並行に設定した通過ラインを横切る様子を分析することで、通過した人数をカウントし、併せて乗車・下車の方向を決定する方式である。本方式は以下の 4 段階の処理から構成される。

##### (1) フレーム間差分と二値化処理

動画の各フレームについて、現フレームと前フレームの差分を取り、その結果を白黒二値化処理する。この処理により、車両内の壁や床、さらに乗車したままの人の部分は差分値 = 0 のため黒色となり、動いている人物部分が白色となる (図 2)

##### (2) グラフ作成処理

ドアと並行に 2 本の通過ライン (A 線, B 線) を設定する。それぞれの通過ライン上の白画素数をカウントし、時間軸に沿ってグラフ化する (図 3 青線)。

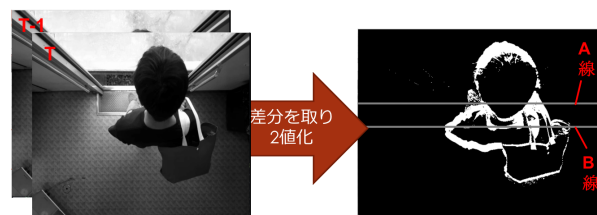


図 2 フレーム間差分と 2 値化処理

##### (3) 平滑化処理

このままではノイズが多いので、ガウシアンフィルタを用いて平滑化を行う (図 3 オレンジ線)[3]

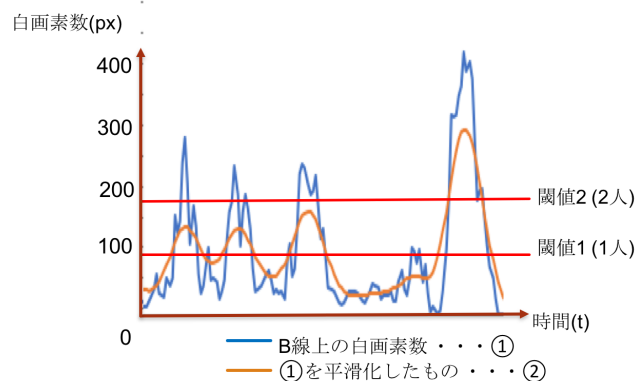


図 3 白画素数を時間軸上で表したグラフ

##### (4) カウント処理

B 線上の白画素数の値のピークが、閾値 1 ~ 閾値 2 までの間の場合は 1 人、閾値 2 を越える場合は 2 人が乗降したとしてカウントする。

また、乗車か下車かの判断は、B 線のグラフがピークの場所で、A 線のグラフが増加傾向にあれば下車、減少傾向にあれば乗車として判断できる。

なお、動画のフレームレート (fps) は 24fps、画像サイズは 640x480 である。

#### 3.3 従来システムの問題点

フレーム間差分方式であるが問題点が存在する。前述のように、B 線上の白画素数を平滑化した値のみを用いてカウントの判断をしている、つまり特徴量を 1 つしか用いていないため、1 人・2 人の閾値をどんなに最適な値に設定しても、誤認識が発生することである。例えば、大柄な人の通過を 2 人の通過と検出したり、子供など小柄な方を検出できないなどの問題が発生する。

## 4. 提案手法

### 4.1 ニューラルネットワークを用いたカウント精度向上

#### 4.1.1 ニューラルネットワークとは

人間の脳内にある神経細胞（ニューロン）とそのつながりを人工ニューロンという数式的なモデルで表現したものである。入力層でデータを受け取り、その値に重み  $w$  を掛けて  $y$  に結果を出力する（図 4）。

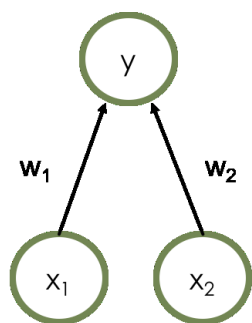


図 4 単純パーセプトロン

一つ一つの人工ニューロンは単純な仕組みだが、これを多数組み合わせる事で複雑な関数近似を行う事ができるのが、ニューラルネットワークの大きな特徴である（図 5）[4]。

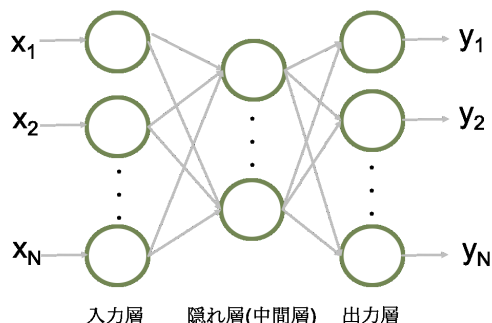


図 5 ニューラルネットワーク

#### 4.1.2 ニューラルネットワークを用いた従来システムの問題点の解決案

フレーム間差分方式においては、特徴量を 1 つしか用いていないため 1 人・2 人を判別する閾値をどんなに最適な値に設定しても、誤認識が発生していた（図 6）。

そこで、特徴量を増やすことによって誤認識を減らす検討を行った。使用する特徴量は図 7 上段のようなグラフが理想的だが、実際は下段のようにノイズが入る。そこで、ガウシアンフィルタを用いて平滑化を行い、理想値に近似している。しかし、人の通り方や人数によって最適な平滑化度合い（図 8）は異なる。

そこで、多重スケールスペースという考え方を導入する。多重スケールスペースは、ガウシアンフィルタの平滑

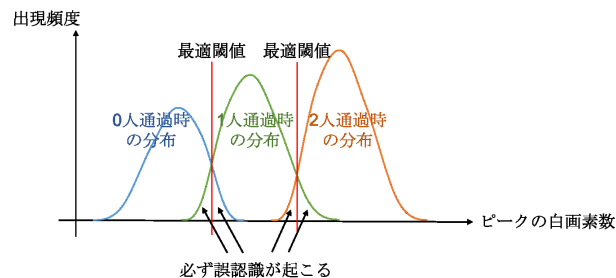


図 6 特徴量分布の問題

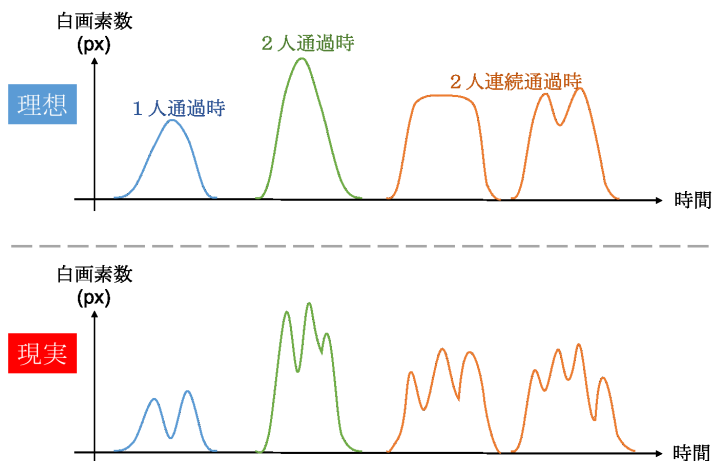


図 7 理想的なグラフ（上段）と実際のグラフ（下段）

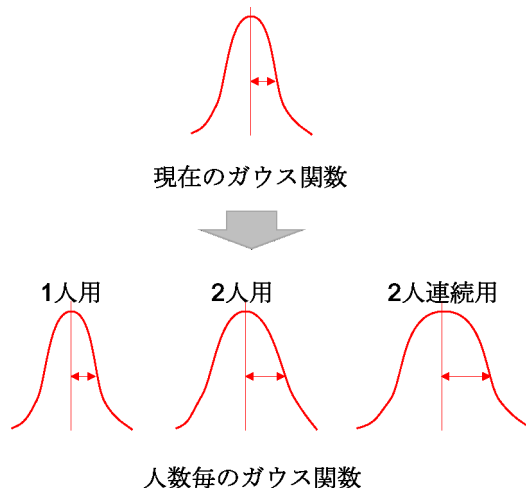


図 8 人数・通り方毎のガウス関数

化度合いであるスケールを大小さまざまな値で入力することで、幅広い特徴（荒い（大局的な）特徴から高精度の（局所的な）特徴）を捉えることができるようになる。よってこれまでは平滑化後のグラフ 1 本を特徴量として用いていたが、今回はスケールの異なる平滑化度合いのグラフ 3 本を用いることとした（図 9）。

図 9 の様に 1 人・2 人通過時の比較を行うと、通過人数によってグラフの形状が異なっていることがわかる。通過人数それぞれにおけるグラフの形状を学習させるこ

とによって、0人・1人・2人通過を推定できるニューラルネットワークを構築する。学習データにはグラフの頂点とその前後フレームの値を用いた。以降、この推定方式を「フレーム間差分+NN方式」と表記する。

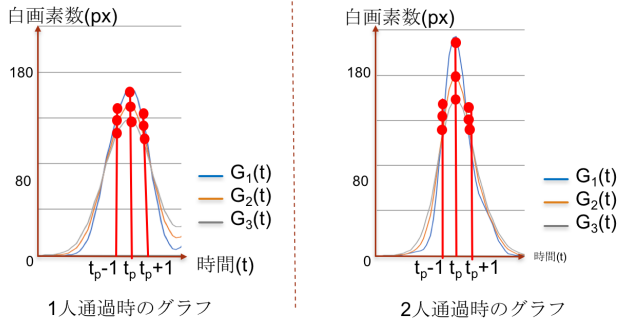


図 9 白画素数を 3 つのスケールで平滑化したグラフにおける 1 人・2 人通過時の比較

## 5. 評価実験

### 5.1 使用ライブラリ

ニューラルネットワークの実現には下記のライブラリを使用した。

TensorFlow[5]

TensorFlow とは Google の開発した機械学習に用いるためのオープンソースライブラリである。

Keras[6]

Keras とは DeepLearning 用のラッパーライブラリである。TensorFlow を扱いやすくするために使用した。

### 5.2 データセットの作成

はじめに、ニューラルネットワークに入力する学習データと教師データのセット (以下、データセット) を作成する。

学習データには、白画素数を異なるスケールで平滑化した各グラフのピーク (図 10) とその前後フレームの値を入力として用いる。

この学習データに人数の情報を教師データとして持たせ、データセットを作成する (表 1)。

表 1 学習データと教師データのデータセット

| フレーム番号 | 学習データ    |          |          | 教師データ                |
|--------|----------|----------|----------|----------------------|
|        | G1       | G2       | G3       | 正解人数                 |
| tp - 1 | G1(tp-1) | G2(tp-1) | G3(tp-1) | 0 人 or 1 人<br>or 2 人 |
| tp     | G1(tp)   | G2(tp)   | G3(tp)   |                      |
| tp + 1 | G1(tp+1) | G2(tp+1) | G3(tp+1) |                      |

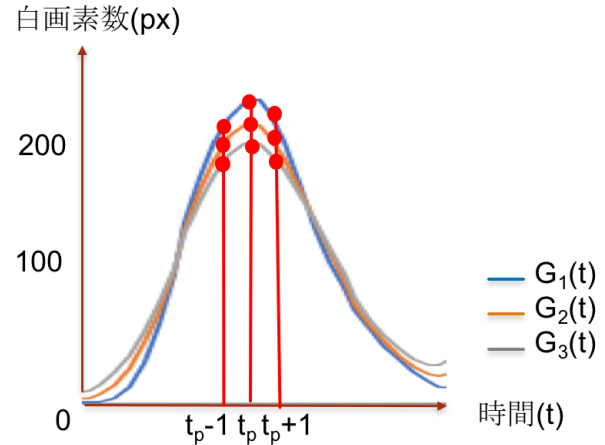


図 10 白画素数を異なる 3 つのスケールで平滑化したグラフ

今回の実験では平成 29 年 8 月の平成筑豊鉄道実験時に撮影した動画 21 本のうち、最も人の乗降の多い 6 本の動画を選び、この動画を 3 本ずつセット 1・セット 2 に分けてデータセットを作成した (表 2)。あわせて 1 セットあたりのサンプル数 (ピーク部の数) を示す。

表 2 データセット作成に用いた動画

| 使用動画 | データセット | サンプル数 |
|------|--------|-------|
| 動画 1 | セット 1  | 154   |
| 動画 2 |        |       |
| 動画 3 |        |       |
| 動画 4 | セット 2  | 164   |
| 動画 5 |        |       |
| 動画 6 |        |       |

### 5.3 学習

9 個の入力、32 個の中間層、3 個の出力を持つニューラルネットワークを作成し、前項にて作成したデータセットを学習させる (図 11)。図 11 のデータセット内の表の数値は例である。

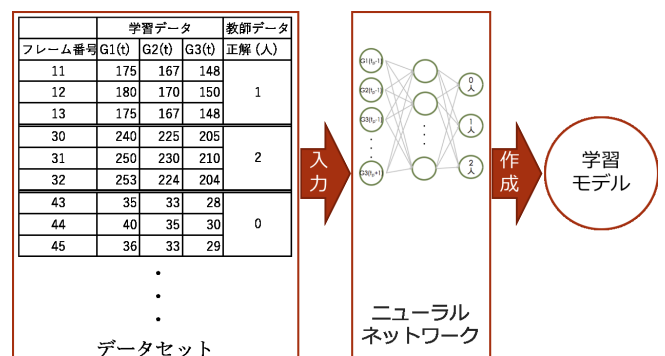


図 11 ニューラルネットワークの学習

また、学習に用いた各パラメータを表 3 に示す。

表 3 各パラメータ

| 学習アルゴリズム     | 誤差逆伝播法              |
|--------------|---------------------|
| 中間層数         | 32                  |
| 学習回数 (epoch) | 20000               |
| 勾配法          | Adam(学習率初期値:0.0001) |
| 活性化関数 (中間層)  | relu                |
| 活性化関数 (出力層)  | softmax             |

#### 5.4 認識

前項にて作成された学習モデルを用いることによって、白画素数を異なるスケールで平滑化した値の各グラフのピークとその前後フレームの値を入力とすることにより、出力として人数が推定できる (図 12)。

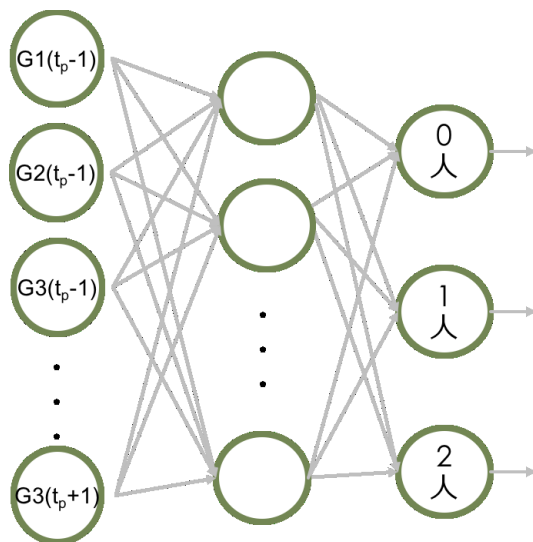


図 12 構築したニューラルネットワーク

#### 5.5 実験結果

各指標の説明 (表 4) と、各動画における精度 (表 5) は次に示す通りである。

表 4 各指標の説明

|      |                                                  |
|------|--------------------------------------------------|
| 再現率  | 実際の乗降した客のうち、システムが正しくカウントできた数の割合 (再現率 = 1 - 未検出率) |
| 未検出率 | 実際に乗降したが、システムがカウントしなかった数の割合                      |
| 過検出率 | 実際には乗降しなくてカウントしないはずのところを、システムがカウントした数の割合         |
| 適合率  | 乗降したとしてカウントされたものの中で、正しかった数の割合 (適合率 = 1 - 誤検出率)   |
| F 値  | 再現率と適合率の調和平均                                     |

旧方式では 82.0% であった精度に対して、フレーム間差分+NN 方式では 90.5% と、8 ポイント程の精度向上を達成できた。

表 5 各動画における精度 (F 値) の値

| 学習     | 認識   |       |        |       |          |
|--------|------|-------|--------|-------|----------|
| データセット | 認識動画 | 再現率   | 適合率    | F 値   | F 値 (総合) |
| セット 1  | 動画 4 | 85.5% | 93.6%  | 89.3% | 90.5%    |
|        | 動画 5 | 84.1% | 97.1%  | 90.1% |          |
|        | 動画 6 | 82.1% | 89.3%  | 85.6% |          |
| セット 2  | 動画 1 | 97.8% | 92.1%  | 94.9% | 90.5%    |
|        | 動画 2 | 82.9% | 94.3%  | 88.2% |          |
|        | 動画 3 | 78.6% | 100.0% | 88.0% |          |

## 6. おわりに

従来方式である、フレーム間差分法を用いたカウントシステムの精度は 7~8 割程であり、実証実験先の平成筑豊鉄道より要求されている最低 8 割という精度を十分に満たしていなかった。そこで新たにニューラルネットワークを用いてフレーム間差分によって抽出した人物の特徴を学習・認識させる方法を提案し、カウント精度の向上を試みた。結果として、H29 年 8 月に平成筑豊鉄道にて人物の乗降の様子を撮影した動画を用いて実験を行った結果、旧方式では 82.0 % であった精度に対してフレーム間差分+NN 方式では 90.5 % と、8 ポイント程の精度向上を達成できた。

今後の課題として、NN の各パラメータの最適化やリカレントネットワークを採用することによるカウント精度の向上を図ることが挙げられる。また、ラズベリーパイという制約下で NN が安定して動作するか検証を行って行く必要があると考えている。

## 謝辞

本研究を行うにあたり、実証実験の環境を提供いただいた平成筑豊鉄道の関係者の皆様、研究に際し協力いただいた研究室の皆様、その他協力いただきました関係者の方々に深く感謝いたします。

## 参考文献

- [1] Raspberry Pi - Teach, Learn, and Make with Raspberry Pi, <https://www.raspberrypi.org/>
- [2] ローカル鉄道向け乗降者カウントシステムの開発, 第 15 回 ITS シンポジウム 1-B-09、2017 年 12 月, 廣重法道, 他
- [3] Scaling Theorems for Zero Crossings, ALAN L. YUILLE and TOMASO A. POGGIO, 1986
- [4] ニューラルネットワークの基礎解説: 仕組みや機械学習・ディープラーニングとの関係は | ビジネス+IT, <https://www.sbb.it.jp/article/cont1/33345>
- [5] TensorFlow, An open-source software library for Machine Intelligence, <https://www.tensorflow.org/>
- [6] Keras: The Python Deep Learning library, <https://keras.io/>