令和2年度卒業論文

画像処理を用いた立石港待機レーンの

混雑量計測システムの開発

令和 3年 1月 14日

学籍番号：163038

氏名：宮地香樹

指導教員：田房　友典

弓削商船高等専門学校情報工学科

目次

[1. はじめに 3](#_Toc63801432)

[2. 概要 4](#_Toc63801433)

[2.1 システム全体の構成 4](#_Toc63801434)

[2.2 開発環境 4](#_Toc63801435)

[3. カメラについて 5](#_Toc63801436)

[3.1 使用するカメラと映像について 5](#_Toc63801437)

[3.2 カメラの設定 6](#_Toc63801438)

[4. 車両検出処理について 7](#_Toc63801439)

[4.1 画像処理の流れ 7](#_Toc63801440)

[4.2 ディレクトリの監視 8](#_Toc63801441)

[4.3 特徴点検出 8](#_Toc63801442)

[4.4 背景差分・エッジ検出 9](#_Toc63801443)

[4.5 車の台数計測 10](#_Toc63801444)

[5. 動作実験 11](#_Toc63801445)

[5.1 実験内容 11](#_Toc63801446)

[5.2 結果 11](#_Toc63801447)

[5.3 誤差 12](#_Toc63801448)

[5.4 改善案 12](#_Toc63801449)

[6. Webの表示部 13](#_Toc63801450)

[7. おわりに 14](#_Toc63801451)

[8. 参考文献 15](#_Toc63801452)

1. はじめに

現在，上島町には他の島につながる橋がなく，陸路での移動は不可能である．そのため，住民は通勤や通学などに船を利用して生活を送っている．その中で車を伴う移動の際に最も利用されているのが，立石港と長崎桟橋をつなぐ生名フェリーである．利用客が多い要因として，航路が短く，乗船時間，待ち時間共に短いことが挙げられる．

一方で，利用客が多いことによる問題も発生している．それは，朝方の通勤ラッシュや，夕方の帰宅ラッシュの時間帯の車の量が多く，待機レーンが埋まってしまうことである．さらに2021年には，岩城橋の開通が予定されている．岩城橋が開通することで，従来利用していた航路より，生名フェリーを利用したほうが，移動時間が短縮され利便性が高まる．これによって生名フェリーの利用者がさらに増加すると予想されている．対策として，より多くの車両を運搬できる新しいフェリーを導入し，立石港の待機レーンを1つ増加させた．

本研究では，上記の対策に加えて，ソフトウェア面での支援を目的とし，待機レーンの混雑状況を住民が把握できる仕組みを作り，さらに混雑を解消させるシステムを開発する．

現在，車量を計測するシステムには，Deep Learning アルゴリズムであるYOLOを用いて物体検出を行う手法[1][2]や，カメラの映像から路面との幾何学変換を行い自動車の位置を決定する方法[3]がある．しかし，これらの手法は交差点や直線道路などの車が通過する際の車両検知に用いられており，待機レーンのような車が静止した時間が多い場面には適していない．また，車両の大部分が隠れてしまった場合には，車両を正しく検知できない可能性もある．さらに，待機レーンには駐車場のように車の止まる場所が定められていないため，トラックなどの大型車を考慮すると車の台数だけでは実際の待機レーンの混雑量を求められない．そこで本研究では待機レーンの正面側にカメラを設置し，背景差分とエッジ検出を用いた画像処理によって車両を検知し，待機レーンの混雑量を計測する．言語はPython Open-CVを使用する．この計測結果をWeb上で公開し，住民に待機レーンの混雑状況を知らせる．Web上では，視覚的に分かりやすくするために，棒グラフを用いて待機レーンの混雑量を表す．Web画面作成に使用する言語はHTML，PHP，JavaScript，MySQLである．

1. 概要
   1. システム全体の構成

　本システムの全体の構成は図2.1のようになっている．IPカメラを待機レーンの正面に設置し，待機レーンを図の左の画像のような角度で一定時間おきに撮影する．撮影した画像をサーバ上に転送し，その画像を使用して画像処理を行う．このカメラとサーバはローカルなネットワークで構成されている．画像処理で得た待機レーンのデータをクラウド上のWebサーバに送る．このデータを棒グラフでWeb上に公開し，住民が混雑度を把握することができる．画像の転送をローカルなネットワークで行い，グローバルなネットワークではデータのやり取りのみを行うことで最低限のセキュリティを確保している．

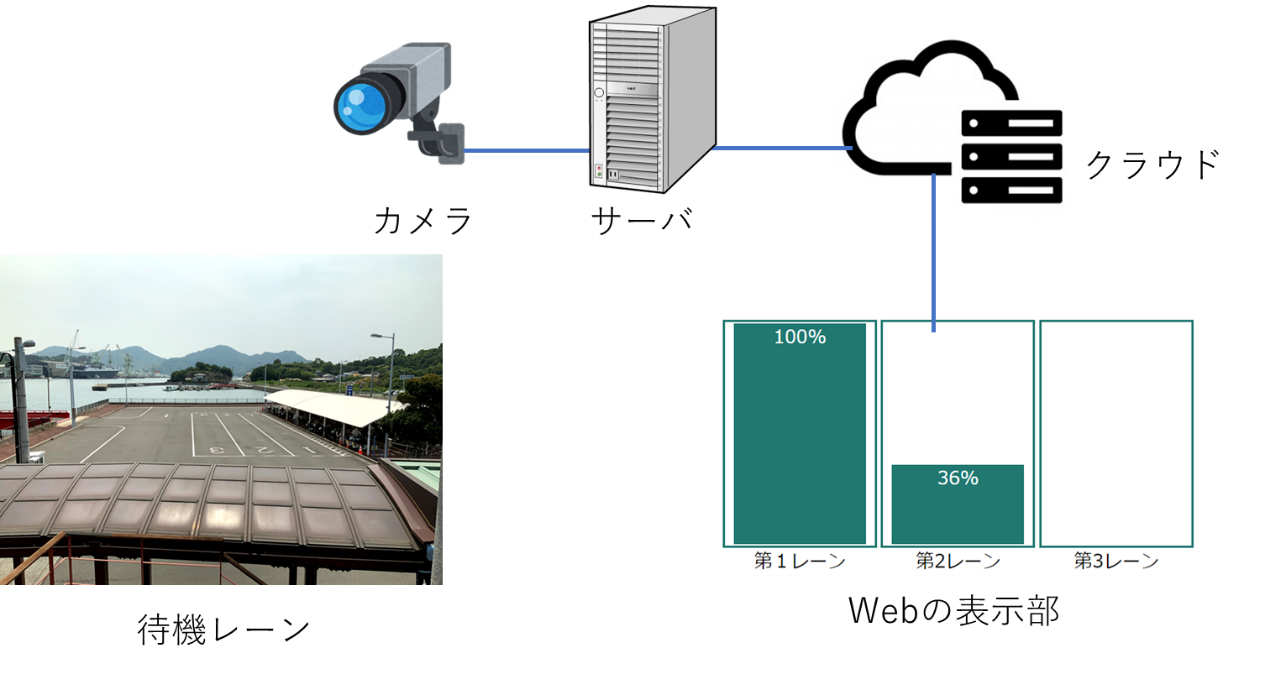


図2.1 システム構成

* 1. 開発環境

　開発に使用した言語と，開発に使用したソフト・環境は以下の通りである．

言語：Python Open-CV，PHP，HTML，JavaScript，MySQL

環境：anaconda，Ubuntu，Visual Studio Code

1. カメラについて
   1. 使用するカメラと映像について

　使用するカメラはPanasonicのi-Pro WV-S1511LN（図3.1）である．このIPカメラには以下のような機能が備わっている．下記の機能を利用して，待機レーンを一定時間おきに撮影し，その画像を転送する．

* スケジュール管理機能
* 一定時間おきの画像撮影（1s/3s/5s/10s/15s/30s/60s…　この中から設定可能）
* 画像の転送
* 認証設定

本研究では，実際に待機レーンに設置するものと同じカメラを用意し，カメラの設定方法の確認と簡単な動作実験を行う．実験に使用している待機レーンの映像は，指導教員の協力のもと別カメラで撮影したものである．理由は，コロナウィルスの関係で学校に登校することが出来ず，現地に赴くことも困難であり，いつ再開するかも不明な状態だったためである．



図3.1　使用するカメラ（i-Pro WV-S1511LN）

* 1. カメラの設定

具体的に行った設定と方法について説明する．まずは付属のセットアップCD-ROMからEasy Configディレクトリに存在する簡単IPセットアップを起動する．配線を正しく行っていると図3.1のように，使用するカメラが表示される．カメラに接続して認証情報を追加し，アクセス権限を設定する．次にスケジュール管理と一定時間おきの画像撮影の設定を行う．認証情報を入力すると，カメラからのストリーミング映像が表示される．設定をクリックすると設定画面へ遷移する．図3.1.2がスケジュール管理の設定画面である．図のように動作モードと動作する時間帯を設定することができる．今回は，画像を送信するため動作モードをFTP定期送信とし，スケジュールは船の稼働時間に合わせて設定を行った．

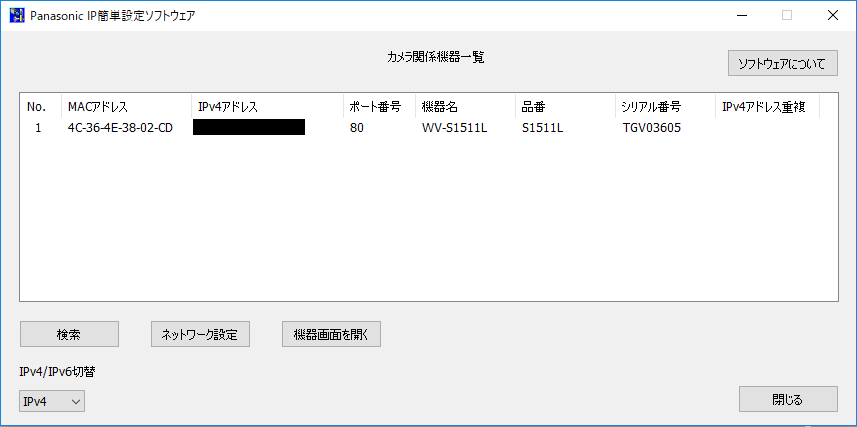


図3.1.1 簡単IPセットアップの起動画面



図3.1.2 スケジュール設定画面

1. 車両検出処理について
   1. 画像処理の流れ

　画像処理の流れを図4.1に示す．ディレクトリの監視を行い，画像が送られてきた場合にのみプログラムが動作する．画像が送られてきたら，車が一台もない待機レーンの画像と，撮影された待機レーンの画像を用いて背景差分を行う．図4.2が，車が止まっていない待機レーンの画像と，撮影した待機レーンの画像の一例である．背景差分を行う際に，より精度を上げるために特徴点検出を行い，撮影した画像の位置を修正する．背景差分を行ったあとにエッジ検出を行う．指定した範囲の白画素を計測して，車がそこにあるかどうかを判定する．

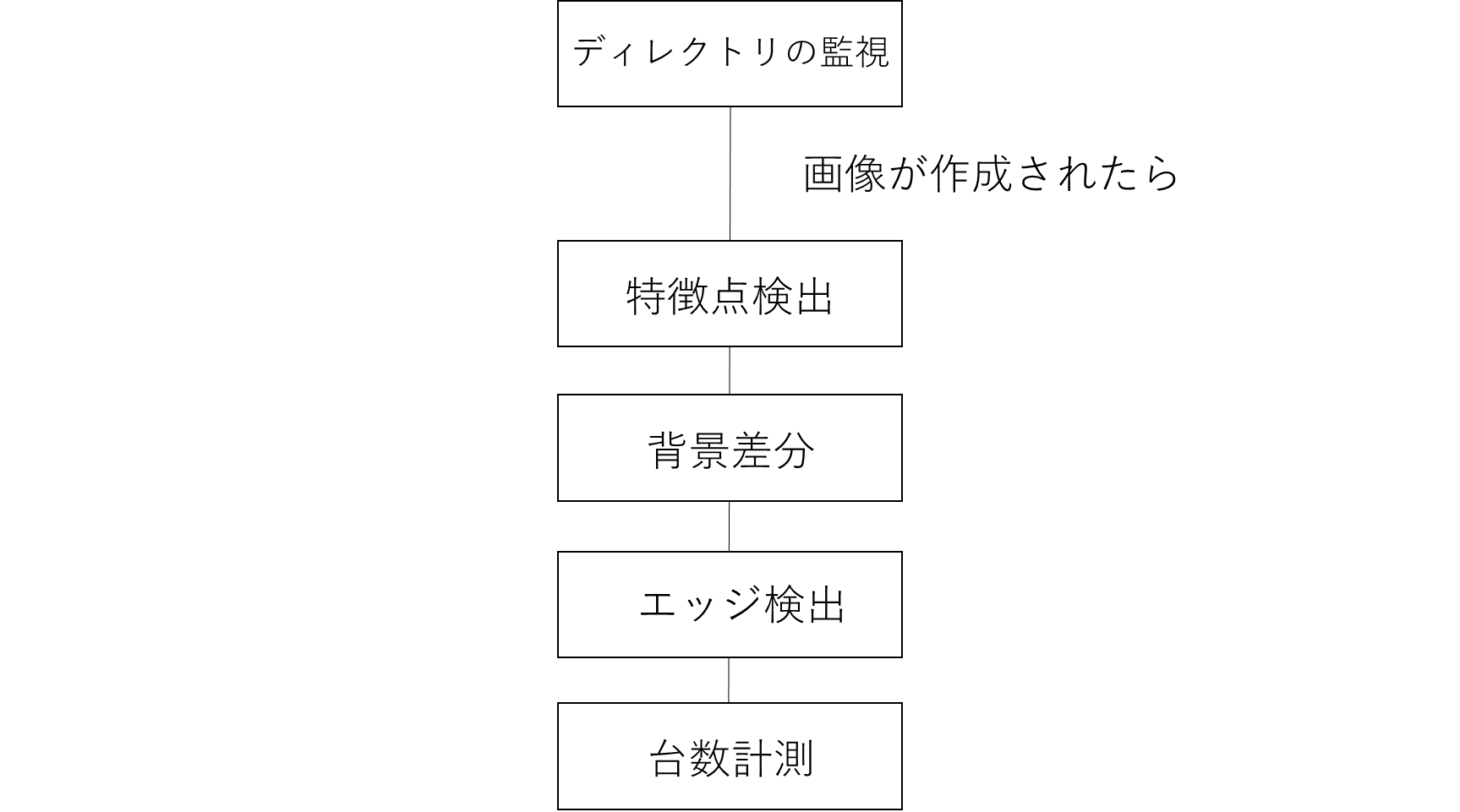


図4.1　画像処理の流れ



図4.2　車のない待機レーン画像と撮影した待機レーン画像の例

* 1. ディレクトリの監視

　ディレクトリの監視にはPythonのwatchdog[4]というライブラリを使用している．このwatchdogは，任意のディレクトリの任意のパターンの名前を持つファイルのイベントを監視し，特定のイベントが発生した場合に，任意の作業を行うようにするモジュールである．特定のイベントには，ファイルの作成や変更，削除などがある．任意のパターンの名前は，正規表現を利用して拡張子を指定したり，ファイルの先頭の文字を指定したりすることができる．今回作成したプログラムでは，イベントにファイルの作成を指定し，名前のパターンとして拡張子にPNGを指定することで，画像ファイルが作成されたときに動作する仕組みを実現している．

* 1. 特徴点検出

　カメラは野外に設置するため，風などの自然現象により多少角度がずれることが予測される．しかし，後述する背景差分を利用するには，できる限り2つの画像が車以外は同じ条件であることが望ましい．そこで，2つの画像に特徴点検出[5]を行い，画像の位置を修正する．設定する特徴点は，図4.3に示すような待機レーンの前側である．待機レーンの前側は車が停車した場合でも遮られることがなく特徴点として適している．これによって，撮影した画像と用意した画像に多少のずれが生じた場合でも，位置を調整し正しく背景差分を行うことが可能になる．



図4.3　特徴点検出を行う場所

* 1. 背景差分・エッジ検出

　車の位置を検出するには背景差分とエッジ検出を用いる．背景差分とは，少し前の状態の背景画像と，物体の入り込んだ画像を比較し，2つの画像の差をとることで物体を検出する手法である．特徴点検出によって位置を修正した画像と，車が一台も止まっていない画像の2つを用いて背景差分[6]を行い，車を検出する．実際に背景差分を行った画像が図4.4.1に示す．この画像にエッジ検出[7][8]を行う． エッジ検出とは，物体の形や材質，色が大きく変わるエッジ部分を検出することである．このエッジ検出を行うことによって，物体の輪郭を求めることができる．エッジ検出を行う理由は，背景差分のみを行うよりも精度が高くなったためである． 図4.4.2は背景差分のみを行って計測した場合と，背景差分とエッジ検出を両方行った場合での誤差の比較である．背景差分のみを行った場合では，車両の数が少ない，または多くなるほど誤差が大きい傾向にある．この結果から，本研究では背景差分とエッジ検出を使用して，車両の検出を行う．

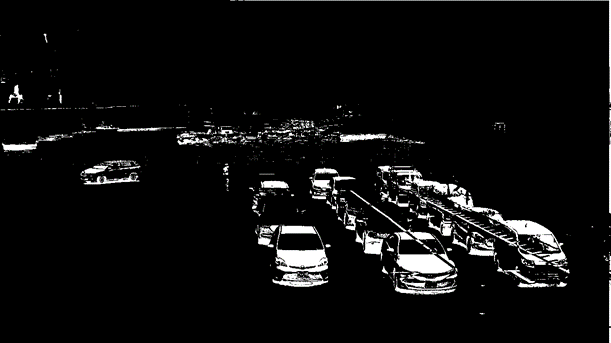


図4.4.1　背景差分を行った画像

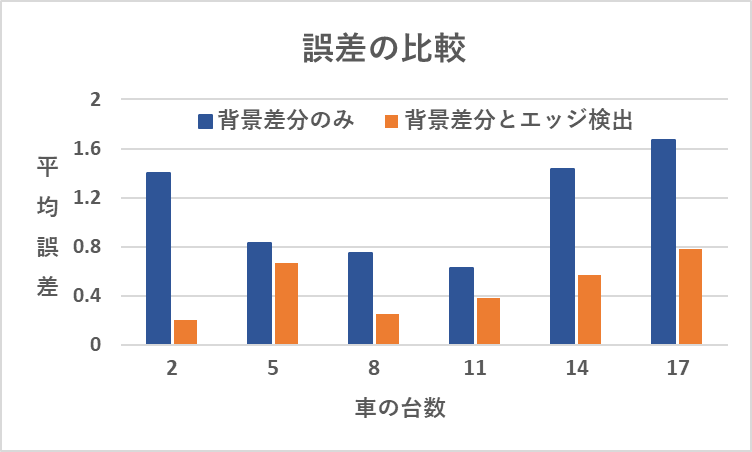


図4.4.2　背景差分のみとエッジ検出も利用した場合の誤差比較

* 1. 車の台数計測

　車の台数の計測は，指定した範囲の白画素数を計測する[9]ことによって行う．図4.5のようにエッジ検出を行った画像のうち，車がいると想定される範囲を矩形で囲む．囲まれた矩形領域に存在する白画素数を測定し，一定以上の白画素数があった場合にはそこに車がいると判定している．車があると判定された回数によって車の台数を求め，それぞれのレーンごとに数列に格納する．

それぞれのレーンの車の台数から，レーンの埋まり具合を求める．レーンの埋まり具合については1つのレーンにおよそ6台の車が入ると想定し，求めたレーンごとの車の台数を6で割って求める．これを百分率で表すために，100でかけてレーンの埋まり具合とする．車の台数だけでなく，レーンの埋まり具合を求める理由は，車の台数よりも，レーンの埋まり具合のほうが，船に乗れるかどうかにおいて重要だからである．生名フェリーの利用客は，一般的な乗用車ではない場合も多くある．例えば，トラックなどの大型車の場合は1台であっても，2台以上の幅をとることがある．こういった場合に，車の台数ではなくレーンの埋まり具合を求めた方が，誤差が小さくなる．

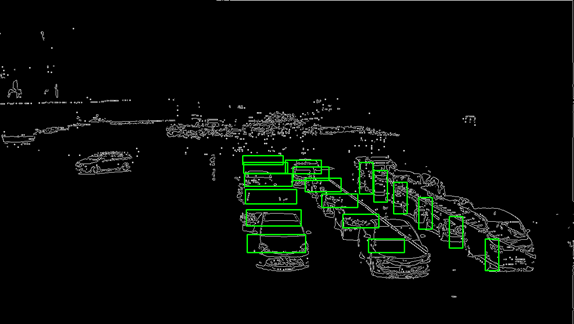


図4.5　車の台数計測を行う際の指定範囲

1. 動作実験
   1. 実験内容

　作成した画像処理プログラムの動作実験を行う．それぞれ違う状態の待機レーンの画像を51枚用意する．ここで利用する画像は，前述した別カメラで撮影した映像から，車の台数が大きく変化した場合に，画像として保存したものである．この画像を，カメラから転送されてくることを想定し，手動で指定のディレクトリへ移動させる．ディレクトリに画像ができたかを判断し，プログラムが正しく動作するかを調べる．またプログラムが待機レーンの混雑状況をどれほど正確に計測できているかを調べるために，出力結果と実際の待機レーンの状態を比較し，誤差を求める．

* 1. 結果

　実験の結果，画像処理プログラムは 51枚すべてにおいてエラーなく出力結果を返すことが確認できた．このことから，本システムは，画像が指定ディレクトリに保存された場合に正しく動作することが分かる．今回行った実験では，手動で画像の保存を行っているが，指定のディレクトリに画像を保存することができれば，プログラムは動作する．そのため，カメラからの転送であっても，画像を指定ディレクトリに保存すれば，プログラムは正しく動作することが可能である．

実際にカメラからの転送を行う場合には今回のように50枚などの量では収まらない．そこで，画像ファイルが増えすぎて重くならないようにするために，画像ファイルを同じ名前にして上書き保存を行い，再度実験を行おうとしたが，プログラムが正しく動作しなかった．これは，watchdogライブラリが，画像が作成された時に動作するよう設定されているからだと考える．上書き保存では，ファイルは作成ではなく変更という形になる．そのため，現在のプログラムでエラーなく動作させるためには，画像の名前はユニークなもので新規保存する必要がある．ユニークな名前については，カメラから転送する際に日時をファイル名にすることが可能であるため，解決することができる．ストレージの容量については，一定期間おきに画像を削除するプログラムを作成することによって対策する．

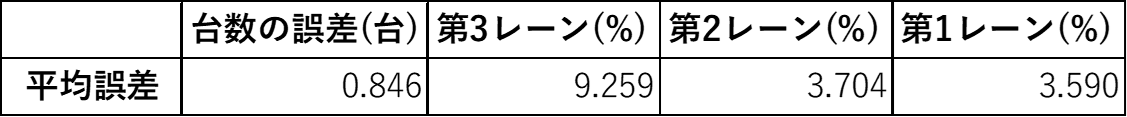
* 1. 誤差

　出力結果と実際の待機レーンの状態を比較する．比較は，台数とレーンの埋まり具合に分ける．実際の待機レーンの埋まり具合は，式①のように求める．Nsが現在停まっている車の台数を，Fsが実際に停まることができた台数を表している．実際に止まることができた台数は，画像を保存する際に動画の時間を進めて調べ，記録したものを使用している．

現在の待機レーンの埋まり具合＝　Ns / Fs \* 100　　　　…式①

例として車が4台停まっており，最終的には6台まで停まることができた場合には66.7%とする．同じく車が4台停まっていた場合でも，最終的に車が5台しか止まらなかった場合には80%として計算している．この結果と画像処理の出力結果の差を絶対値で表したものを誤差とする．その結果，51枚の画像の平均誤差は表5.3のようになった．この表から，第1レーンと第2レーンの誤差は5%未満であるのに対し，第3レーンの誤差は約10%と，第1・第2レーンに比べて誤差が大きいことが分かる．これは，第3レーンがカメラから見て正面に位置しているため，車間距離や車の高さなどの影響を受けやすいからだと考えられる．

表5.3　画像処理結果と実際の待機レーンの誤差



* 1. 改善案

　誤差の改善案として挙げられるのはカメラの位置の変更である．前述したとおり，第3レーンの誤差が大きい要因として挙げられるのが，カメラとの位置関係である．そこで，カメラの位置を本研究で設置した場所から，より左側に設置する．これによって，カメラから見た第3レーンの位置が，他のレーンと同じように斜めから見た位置関係になり，誤差が減少すると考える．

もう一つの案としてプログラムの修正が挙げられる．現在のプログラムは台数も計測するために，1つのレーンに白画素を測定する領域を6つに設定している．しかし，待機レーンで重要なのは台数ではなくレーンの埋まり具合である．そこで，台数の計測を断念し，白画素を測定する領域をより細かく細分化する．これによって，より正確に待機レーンがどこまで埋まっているかを判断することができる．

1. Webの表示部

画像処理で求めた結果をWeb上に表示することで，住民が待機レーンの混雑度を把握することが出来る．Webページの作成にはHTMLとCSSを利用しており，データの受信にPHP，データベースへの接続にMySQLを使用している．図6が作成したWebページの待機レーンを表示する部分である．図のように数値だけでなく棒グラフを用いて視覚的にわかりやすく表現している．画像処理の結果はjson形式でWeb上のサーバに送信される．これを読み取り，図のようなグラフを作成している．またWebページには，生名フェリーの時刻表と，現在の時刻を表示している．これを掲載することにより，船がいつ発着するのかを現在の時刻と照らし合わせて確認することができる．

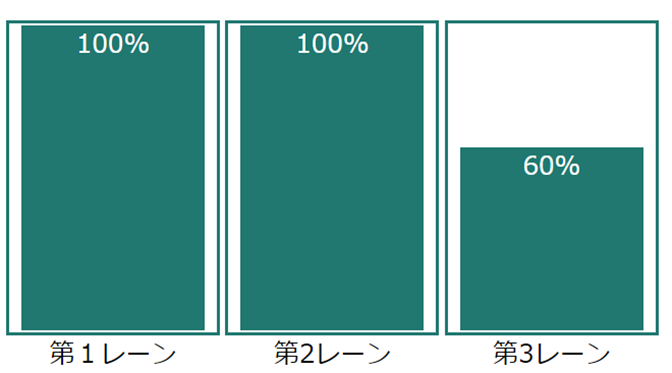


図6　Webで表示される棒グラフ

1. おわりに

　背景差分法とエッジ検出を用いて待機レーンの混雑量を計測することができるシステムを開発した．このデータをWeb上で公開し，住民が待機レーンの混雑具合を把握することができる．また，作成したプログラムの検証を行い，精度を調査した．調査した待機レーンの混雑量の誤差は，第1・第2レーンで5%未満であり，十分実用が可能な範囲である．しかし，第3レーンの誤差は約10%と，他のレーンより高くなった．そこで，誤差が大きい理由について考察を行い，改善案を提案した．

しかし，このシステムにはまだ解決すべき問題がある．それは，このシステムだけでは混雑状況を改善することはできないことである．理由は，利用者が船に乗ろうと思ってから，実際に船に乗るまでに時間差があるからである．現在，上島町の住民が船に乗ろうと思ってから港に到着するまでにかかる時間は，平均すると5分から10分である．この間に，車がさらに止まってしまったり，船が出発してしまい，逆に車の量が減ってしまったりする．つまり，出発するとき見た混雑状況と，港に着くときに見た混雑状況が大きく変わってしまい，混雑状況の信頼性が失われてしまうのである．これでは，混雑を回避することができない．

この問題を解決するためには，利用者が出発しようとしたときの混雑度ではなく，到着するときの混雑度を知らせる必要がある．そこで，今回開発したシステムを利用し，曜日や時間帯ごとの待機レーンの混雑度を記録する．この記録した混雑度を利用し，混雑度を予測することができる仕組みを作る．この仕組みを利用して住民が到着するまでにかかる時間を考慮した予測混雑度を住民に伝えることで，この問題を解決することができる．

1. 参考文献

[1]岩崎洋一郎, 竹原洋志, 岩本祥二郎, 宮田俊彦,倉本俊昌, 北島俊孝, 瀬戸口恵, 物体検出アルゴリズム YOLOv2 を用いた交通量と車両走行軌跡の自動計測, 電気学会全国大会講演論文集, 2018, p. 3-109.

[2]谷育馬, 横井昭, 佐治斉. 幾何学的制約を用いた車両抽出. 画像電子学会誌, 2014, Vol.43, No.4, p. 579-587.

[3]峯岸朋弥, 吉田享子. 物体検知・識別器による自動車交通量調査システムの試作と評価, 専修ネットワーク&インフォメーション, 2020, vol28, p53-58

[4]watchdog:ファイル監視ライブラリ

<https://qh73xebitbucketorg.readthedocs.io/ja/latest/1.Programmings/python/LIB/watchdog/main/>

[5]Pythonによる写真画像の位置合わせ

<https://campkougaku.com/2020/04/15/alignment/>

[6]背景差分で物体抽出(画像比較)

<https://qiita.com/goodboy_max/items/71a3b804d14f961c0d91>

[7]canny法によるエッジ検出

<http://labs.eecs.tottori-u.ac.jp/sd/Member/oyamada/OpenCV/html/py_tutorials/py_imgproc/py_canny/py_canny.html>

[8]エッジ検出

<https://axa.biopapyrus.jp/ia/opencv/edge.html>

[9]Python，OpenCVで二値画像から白と黒の面積比を算出

https://techtech-sorae.com/pythonopencvで二値画像から白と黒の面積比を算出/