

横断歩道の本数・摩耗率推定プログラムの README

1. フォルダ構成

名前	更新日時	種類	サイズ
input	2020/09/29 13:41	ファイルフォルダ	
output	2020/10/05 19:09	ファイルフォルダ	
program	2020/10/05 19:09	ファイルフォルダ	
README.pdf	2020/10/06 11:00	Chrome HTML Do...	233 KB

図 1. フォルダ構成(ルートフォルダ)

ご提供するフォルダ構成は、図 1 の通りとなっております。ルートフォルダに、3 つのフォルダ(‘input’, ‘output’, ‘program’)が存在します。次に、それぞれの詳細についてご説明いたします。

1.1 input フォルダ

名前	更新日時	種類	サイズ
crosswalk	2020/09/29 13:41	ファイルフォルダ	
template	2020/09/29 13:41	ファイルフォルダ	
config.json	2020/10/05 20:05	JSON File	1 KB

図 2. ‘input’フォルダ内の構成

‘input’フォルダ内の構成は、図 2 の通りとなっております。‘input’フォルダ内には、2 つのフォルダ(‘crosswalk’, ‘template’)と 1 つのファイル(‘config.json’)が存在します。それぞれのフォルダについては、以下の通りをお願いいたします。

推定を行う横断歩道画像は、‘crosswalk’内に入れてください。

テンプレートマッチングに使用するテンプレート画像は、‘template’内に入れてください。

また、‘config.json’では、横断歩道検出に利用するハイパーパラメータがまとめてあります。

```
{
  "gamma": 2, (①ガンマ補正のガンマ値)
  "num_morpho_iter": 3, (②モルフォロジー処理の繰り返し回数)
  "s_thresh": 20, (③HSV 色空間における彩度の閾値)
  "dif_white_line": 0.9, (④横断歩道の幅の比較における割合)
  "min_white_line": 0.3, (⑤横断歩道の両端で、最大幅の何割以上とするか)
  "std_white_line": 0.5 (⑥基準線上の白色ピクセル数を基準として、何割以下になる
    まで基準線を上下させるか)
  "save_json": false (⑦パラメータ調整用の座標を出力するかどうか)
  "save_image": false (⑧パラメータ調整用の画像を出力するかどうか)
}
```

① ガンマ補正のガンマ値

値が大きいほど、画像が明るくなり、値が小さいほど画像が暗くなるような値となっております。

② モルフォロジー処理の繰り返し回数

繰り返し回数が多いほど、横断歩道のかすれを改善しますが、横断歩道が連結する恐れがあります。逆に、少ないほど横断歩道が連結する恐れが減りますが、横断歩道のかすれの改善は十分にされない可能性があります。

③ HSV 色空間における彩度の閾値 ($0 \leq s_thresh \leq 100$)

彩度の値は大きいほど、画像の背景領域(横断歩道以外の領域)が含まれやすく、値が小さいほど背景領域の影響を受けなくなりますが、横断歩道領域がうまく取れなくなる恐れがあります。

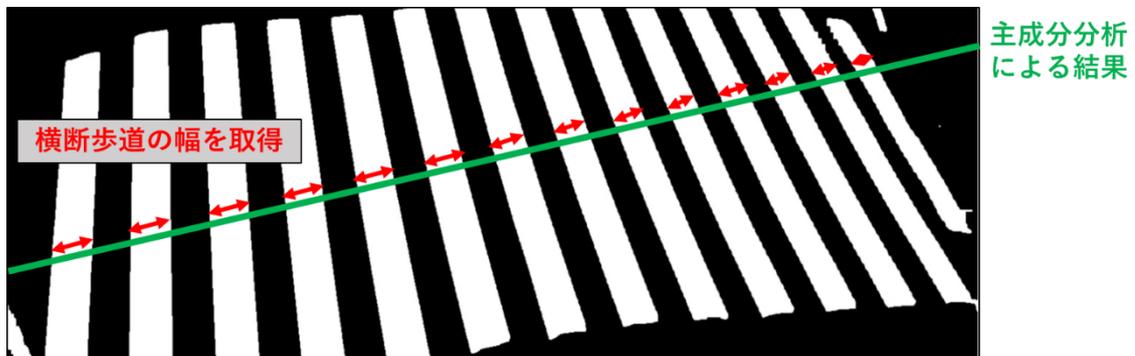


図 3. 横断歩道の白線の幅

④ 横断歩道の幅の比較における割合 ($0 \leq dif_white_line \leq 1$)

横断歩道の幅を降順に並べ替えたものから、大きい順に見ていく際に、この割合を乗算したものと次に大きい幅を比較し、最大幅を求めます。この割合が大きいほど、厳密に横断歩道の最大幅を取得できますが、条件を全て満たさなかった場合は、一番大きい値が横断歩道の最大幅となります。割合が小さいほど、より大雑把に横断歩道の最大幅を取得します。

⑤ 横断歩道の両端で、最大幅の何割以上とするか ($0 \leq min_white_line \leq 1$)

この割合が大きいほど、両端の横断歩道の幅がある程度の大きさでないと横断歩道と認識されなくなり、横断歩道の本数が少なくなります。一方で割合が小さいほど、横断歩道以外の細い線が含まれる可能性があります。

⑥ 基準線上の白色ピクセル数を基準として何割以下になるまで基準線を上下させるか ($0 \leq std_white_line \leq 1$)

この割合が小さいほど、広く横断歩道領域を抽出しますが、横断歩道領域以外が含まれる可能性があります。一方で、大きいほど、狭く横断歩道領域を抽出し、横断歩道全体を抽出できない可能性があります(図 4 参照)。



横断歩道領域抽出結果
(std_white_line=0.1)



横断歩道領域抽出結果
(std_white_line=0.9)

図 4. 基準線の上下の割合

- ⑦ パラメータ調整用の座標を出力するかどうか(true or false)
パラメータ調整のために、座標を出力したい場合は、true に、出力しない場合は、false に設定してください。
- ⑧ パラメータ調整用の画像を出力するかどうか(true or false)
パラメータ調整のために、画像を出力したい場合は、true に、出力しない場合は、false に設定してください。

1.2 program フォルダ

名前	更新日時	種類	サイズ
python	2020/10/06 10:35	ファイル フォルダ	
main.exe	2020/10/05 20:38	アプリケーション	69,957 KB

図 5. 'program'フォルダ内の構成

'program'フォルダ内の構成は、図 5 の通りとなっております。'program'フォルダ内には、1つのフォルダ('python')と1つの exe ファイル('main.exe')が存在します。python フォルダには、exe 化する前の python 形式のプログラムファイルが入っております。

1.3 output フォルダ

'output'フォルダ内には、exe ファイルを実行すると、'results.txt'という名前の本数・摩耗率推定結果が出力されます。

また、'config.json'内の"save"が true の場合は、横断歩道の本数・摩耗率の推定を行う過程の画像がそれぞれ出力されます。

"save_json"や"save_image"が true の場合、ハイパーパラメータ調整用の座標・画像がフォルダ別に保存されます(図 6 参照)。

名前	更新日時	種類	サイズ
img_detected	2020/11/25 17:45	ファイルフォルダー	
img_detected(after)	2020/11/25 17:45	ファイルフォルダー	
img_gray	2020/11/25 17:45	ファイルフォルダー	
img_hsv	2020/11/25 17:45	ファイルフォルダー	
img_PCA	2020/11/25 17:45	ファイルフォルダー	
img_transformed	2020/11/25 17:45	ファイルフォルダー	
img_width	2020/11/25 17:45	ファイルフォルダー	
json_detected(after)	2020/11/25 17:45	ファイルフォルダー	
results.txt	2020/11/25 17:45	TXT ファイル	1 KB

図 6. 'output'フォルダ内の構成("save_json":true, "save_image":true の場合)

1.3.1 座標による出力

json_detected(after):テンプレートマッチングで検出された横断歩道領域に幅を持たせた検出結果.

```

1  [
2      {
3          "x": 0,
4          "y": 705
5      },
6      {
7          "x": 1599,
8          "y": 705
9      },
10     {
11        "x": 1599,
12        "y": 829
13    },
14    {
15        "x": 0,
16        "y": 829
17    }
18 ]

```

図 7. 出力される json ファイルの中身(上から順番に, 左上→右上→右下→左下の座標)

1.3.2 画像による出力

img_detected:テンプレートマッチングで検出された横断歩道領域.

img_detected(after):検出された横断歩道領域に幅を持たせた検出領域.

img_gray:横断歩道領域を二値化した結果.

img_hsv:横断歩道領域を HSV 空間の彩度で閾値処理した結果.

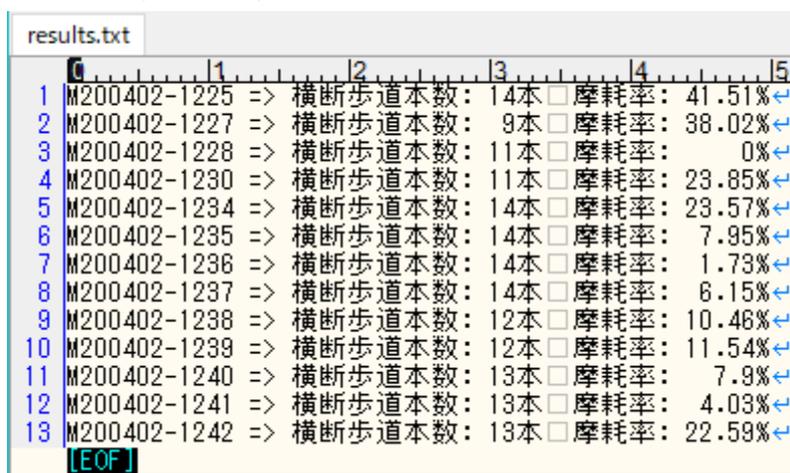
img_PCA:主成分分析による横断歩道の基準線.

img_transformed: テンプレートマッチングによる検出結果を射影変換した結果.

img_width: 横断歩道の左端および右端の検出結果.

2 実行手順

- ① ‘input’フォルダ内の2つのフォルダに横断歩道画像とテンプレート画像があることを確認してください(1.1 input フォルダ 参照).
- ② ‘program’フォルダ内にある, ‘main.exe’ファイルを実行してください.
- ③ ‘output’フォルダ内に, ‘results.txt’というファイルが作成され, そのファイル内に, 推定結果が出力されます(図8 参照).



```
results.txt
0 1 2 3 4 5
1 M200402-1225 => 横断歩道本数: 14本  摩耗率: 41.51% ←
2 M200402-1227 => 横断歩道本数: 9本  摩耗率: 38.02% ←
3 M200402-1228 => 横断歩道本数: 11本  摩耗率: 0% ←
4 M200402-1230 => 横断歩道本数: 11本  摩耗率: 23.85% ←
5 M200402-1234 => 横断歩道本数: 14本  摩耗率: 23.57% ←
6 M200402-1235 => 横断歩道本数: 14本  摩耗率: 7.95% ←
7 M200402-1236 => 横断歩道本数: 14本  摩耗率: 1.73% ←
8 M200402-1237 => 横断歩道本数: 14本  摩耗率: 6.15% ←
9 M200402-1238 => 横断歩道本数: 12本  摩耗率: 10.46% ←
10 M200402-1239 => 横断歩道本数: 12本  摩耗率: 11.54% ←
11 M200402-1240 => 横断歩道本数: 13本  摩耗率: 7.9% ←
12 M200402-1241 => 横断歩道本数: 13本  摩耗率: 4.03% ←
13 M200402-1242 => 横断歩道本数: 13本  摩耗率: 22.59% ←
[EOF]
```

図 8. 本数・摩耗率推定結果 (‘results.txt’)