

## 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横断歩道の白線の幅のずれの許容割合)
  "standard_white_width": 0.3, (⑤横断歩道の両端で、最大幅の何割以上とするか)
  "width_ratio": 0.8, (⑥最大値の幅の何割を横断歩道の幅とみなすか)
  "standard_white": 0.5 (⑦基準線上の白色ピクセル数を基準として、何割以下まで基準線を上下させるか)
}
```

① ガンマ補正のガンマ値

値が大きいくほど、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。

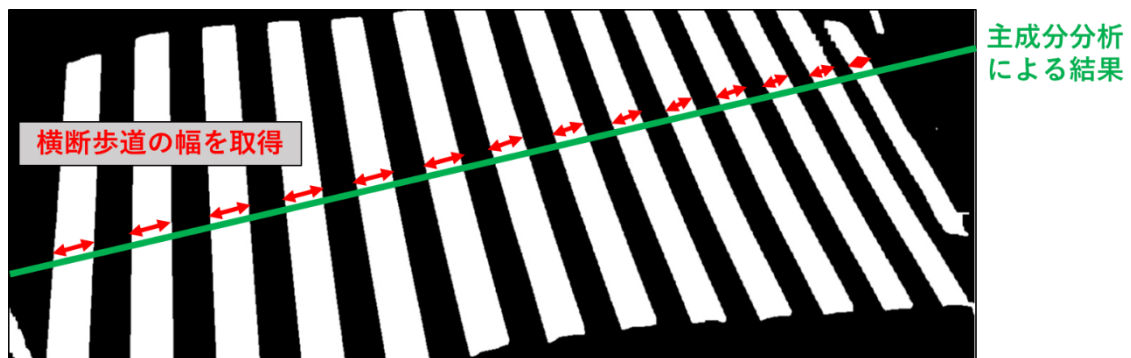


図 3. 横断歩道の白線の幅

④ 横断歩道の白線の幅のずれの許容割合 ( $0 \leq dif\_white\_line \leq 1$ )

許容割合が大きいほど、厳密に横断歩道の幅(図 3 参照)を取得できますが、条件をすべて満たさなかった場合は、最大幅が横断歩道の幅となります。許容割合が小さいほど、より大雑把に横断歩道の幅を取得します。

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

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 最大値の幅の何割を横断歩道の幅とみなすか ( $0 \leq width\_ratio \leq 1$ )

この割合が大きいほど、横断歩道の本数は少なくなり、小さいほど本数が多くなります。

⑦ 基準線上の白色ピクセル数を基準として何割以下まで基準線を上下させるか ( $0 \leq standard\_white \leq 1$ )

この割合が大きいほど、摩耗率が小さく、割合が小さいほど摩耗率が大きくなります。

## 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 4. ‘program’フォルダ内の構成

‘program’フォルダ内の構成は、図 4 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。

## 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 5 参照)。

results.txt	
1	M200402-1225 => 横断歩道本数: 7本 □ 摩耗率: 42.01% ←
2	M200402-1227 => 横断歩道本数: 10本 □ 摩耗率: 36.02% ←
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: 17.87% ←
5	M200402-1234 => 横断歩道本数: 12本 □ 摩耗率: 26.03% ←
6	M200402-1235 => 横断歩道本数: 13本 □ 摩耗率: 4.7% ←
7	M200402-1236 => 横断歩道本数: 14本 □ 摩耗率: 0% ←
8	M200402-1237 => 横断歩道本数: 14本 □ 摩耗率: 2.05% ←
9	M200402-1238 => 横断歩道本数: 12本 □ 摩耗率: 10.35% ←
10	M200402-1239 => 横断歩道本数: 11本 □ 摩耗率: 11.14% ←
11	M200402-1240 => 横断歩道本数: 13本 □ 摩耗率: 2.78% ←
12	M200402-1241 => 横断歩道本数: 13本 □ 摩耗率: 2.23% ←
13	M200402-1242 => 横断歩道本数: 12本 □ 摩耗率: 23.62% ←
	[EOF]

図 5. 本数・摩耗率推定結果 (‘results.txt’)