

# GrassNet

An independent tool to detect grass and its boundary from background

Author: Venus Pagidimarri

Version: 1.0

Release Date: November 24, 2021



<b>1 Namespace Index</b>	<b>1</b>
1.1 Namespace List	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 File Index</b>	<b>5</b>
3.1 File List	5
<b>4 Namespace Documentation</b>	<b>7</b>
4.1 boundary_detection Namespace Reference	7
4.1.1 Detailed Description	8
4.1.2 Variable Documentation	8
4.1.2.1 args	8
4.1.2.2 BLACK	8
4.1.2.3 bound_detect	8
4.1.2.4 cmap	8
4.1.2.5 cnt_no	9
4.1.2.6 constant	9
4.1.2.7 contours	9
4.1.2.8 draw_img	9
4.1.2.9 False	9
4.1.2.10 help	9
4.1.2.11 hierarchy	9
4.1.2.12 hsv	10
4.1.2.13 hue	10
4.1.2.14 hue_binary	10
4.1.2.15 hue_numbers	10
4.1.2.16 image_array	10
4.1.2.17 image_color_pad	10
4.1.2.18 img_dilation	11
4.1.2.19 img_erosion	11
4.1.2.20 int	11
4.1.2.21 kernel_big	11
4.1.2.22 kernel_small	11
4.1.2.23 lists	11
4.1.2.24 pad_size	11
4.1.2.25 parser	12
4.1.2.26 Photos	12
4.1.2.27 required	12

4.1.2.28 res . . . . .	12
4.1.2.29 ret . . . . .	12
4.1.2.30 str . . . . .	12
4.1.2.31 thresh . . . . .	12
4.1.2.32 True . . . . .	13
4.1.2.33 type . . . . .	13
4.2 grass Namespace Reference . . . . .	13
<b>5 Class Documentation</b>	<b>15</b>
5.1 grass.Grass Class Reference . . . . .	15
5.1.1 Detailed Description . . . . .	15
5.1.2 Constructor & Destructor Documentation . . . . .	15
5.1.2.1 __init__() . . . . .	15
5.1.3 Member Function Documentation . . . . .	16
5.1.3.1 contour_len_tresh() . . . . .	16
5.1.3.2 get_max_green_intensity() . . . . .	16
5.1.3.3 get_min_green_intensity() . . . . .	16
5.1.3.4 get_padsizes() . . . . .	16
5.1.3.5 no_dilation_steps() . . . . .	17
5.1.3.6 no_erosion_steps() . . . . .	17
5.1.4 Member Data Documentation . . . . .	17
5.1.4.1 pad_size . . . . .	17
<b>6 File Documentation</b>	<b>19</b>
6.1 detection.py File Reference . . . . .	19
6.2 grass.py File Reference . . . . .	20
<b>Index</b>	<b>21</b>

# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">boundary_detection</a>	7
<a href="#">grass</a>	13



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">grass.Grass</a> . . . . .	15
---------------------------------------	----





# Chapter 3

## File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

<a href="#">boundary_detection.py</a>	19
<a href="#">grass.py</a>	20



## Chapter 4

# Namespace Documentation

### 4.1 boundary\_detection Namespace Reference

#### Variables

- `parser` = argparse.ArgumentParser(description='Detect boundaries of the grass in the given image')
- `type`
- `int`
- `required`
- `help`
- `str`
- `True`
- `False`
- `args` = parser.parse\_args()
- `Photos` = args.file\_dir
- list `lists` = []
- `image_array` = cv2.imread(`Photos` + file\_name)
- `bound_detect` = `Grass`(10)
- `hsv` = cv2.cvtColor(`image_array`, cv2.COLOR\_BGR2HSV)
- `pad_size` = bound\_detect.get\_padsizes()
- list `BLACK` = [0, 0, 0]
- `image_color_pad`
- `hue` = `hsv`[:, :, 0]
- tuple `hue_binary`
- tuple `hue_numbers` = `hue_binary`.astype(np.uint8)
- tuple `constant` = `hue_numbers` \* 255
- `kernel_small` = bound\_detect.no\_dilation\_steps()[0]
- `kernel_big` = bound\_detect.no\_erosion\_steps()[0]
- `img_dilation` = cv2.dilate(`constant`, `kernel_small`, iterations=bound\_detect.no\_dilation\_steps()[1])
- `img_erosion` = cv2.erode(`img_dilation`, `kernel_big`, iterations=bound\_detect.no\_erosion\_steps()[1])
- `ret`
- `thresh`
- `contours`
- `hierarchy`
- `draw_img` = `image_color_pad`.copy()
- `int cnt_no` = 0
- `res` = cv2.drawContours(`draw_img`, `contours`, i, (0, 0, 255), 50)
- `cmap`

### 4.1.1 Detailed Description

Project : GrassNet - An independent tool to detect grass and its boundaries from background.  
Aim : Detect boundaries of the grass in the given image  
Author : Venus Pagidimarri  
Version : 1.0  
Date : November 24th, 2021

Inputs:

1. -pf : Flag to notify if the image is coming from a video or a directory
2. -d : Current directory
3. -i : Image name
4. -rf : Flag to read all images in the directory
5. -df : Flag to display plots
6. -vf : Verbose flag

Output:

1. Image with boundary around grass

### 4.1.2 Variable Documentation

#### 4.1.2.1 args

```
boundary_detection.args = parser.parse_args()
```

#### 4.1.2.2 BLACK

```
list boundary_detection.BLACK = [0, 0, 0]
```

#### 4.1.2.3 bound\_detect

```
boundary_detection.bound_detect = Grass(10)
```

#### 4.1.2.4 cmap

```
boundary_detection.cmap
```

#### 4.1.2.5 cnt\_no

```
int boundary_detection.cnt_no = 0
```

#### 4.1.2.6 constant

```
tuple boundary_detection.constant = hue_numbers * 255
```

#### 4.1.2.7 contours

```
boundary_detection.contours
```

#### 4.1.2.8 draw\_img

```
boundary_detection.draw_img = image_color_pad.copy()
```

#### 4.1.2.9 False

```
boundary_detection.False
```

#### 4.1.2.10 help

```
boundary_detection.help
```

#### 4.1.2.11 hierarchy

```
boundary_detection.hierarchy
```

#### 4.1.2.12 hsv

```
boundary_detection.hsv = cv2.cvtColor(image_array, cv2.COLOR_BGR2HSV)
```

#### 4.1.2.13 hue

```
boundary_detection.hue = hsv[:, :, 0]
```

#### 4.1.2.14 hue\_binary

```
boundary_detection.hue_binary
```

**Initial value:**

```
1 = (hue[:, :] < bound_detect.get_max_green_intensity()) * (  
2     hue[:, :] > bound_detect.get_min_green_intensity())
```

#### 4.1.2.15 hue\_numbers

```
boundary_detection.hue_numbers = hue_binary.astype(np.uint8)
```

#### 4.1.2.16 image\_array

```
boundary_detection.image_array = cv2.imread(Photos + file_name)
```

#### 4.1.2.17 image\_color\_pad

```
boundary_detection.image_color_pad
```

**Initial value:**

```
1 = cv2.copyMakeBorder(image_array, pad_size, pad_size, pad_size, pad_size, cv2.BORDER_CONSTANT,  
2     value=BLACK)
```

#### 4.1.2.18 img\_dilation

```
boundary_detection.img_dilation = cv2.dilate(constant, kernel_small, iterations=bound_detect.no_↵  
dilation_steps()[1])
```

#### 4.1.2.19 img\_erosion

```
boundary_detection.img_erosion = cv2.erode(img_dilation, kernel_big, iterations=bound_detect.no_↵  
erosion_steps()[1])
```

#### 4.1.2.20 int

```
boundary_detection.int
```

#### 4.1.2.21 kernel\_big

```
boundary_detection.kernel_big = bound_detect.no_erosion_steps()[0]
```

#### 4.1.2.22 kernel\_small

```
boundary_detection.kernel_small = bound_detect.no_dilation_steps()[0]
```

#### 4.1.2.23 lists

```
list boundary_detection.lists = []
```

#### 4.1.2.24 pad\_size

```
boundary_detection.pad_size = bound_detect.get_padsizes()
```

#### 4.1.2.25 parser

```
boundary_detection.parser = argparse.ArgumentParser(description='Detect boundaries of the grass in  
the given image')
```

#### 4.1.2.26 Photos

```
boundary_detection.Photos = args.file_dir
```

#### 4.1.2.27 required

```
boundary_detection.required
```

#### 4.1.2.28 res

```
boundary_detection.res = cv2.drawContours(draw\_img, contours, i, (0, 0, 255), 50)
```

#### 4.1.2.29 ret

```
boundary_detection.ret
```

#### 4.1.2.30 str

```
boundary_detection.str
```

#### 4.1.2.31 thresh

```
boundary_detection.thresh
```



#### 4.1.2.32 True

`boundary_detection.True`

#### 4.1.2.33 type

`boundary_detection.type`

## 4.2 grass Namespace Reference

### Classes

- class [Grass](#)



## Chapter 5

# Class Documentation

### 5.1 grass.Grass Class Reference

#### Public Member Functions

- def `__init__` (self, `pad_size`=10)
- def `get_padsizes` (self)
- def `get_min_green_intensity` (self)
- def `get_max_green_intensity` (self)
- def `no_dilation_steps` (self)
- def `no_erosion_steps` (self)
- def `contour_len_tresh` (self)

#### Public Attributes

- `pad_size`

#### 5.1.1 Detailed Description

Grass class is used to maintain the constants and thresholds for grass boundary detection

#### 5.1.2 Constructor & Destructor Documentation

##### 5.1.2.1 `__init__()`

```
def grass.Grass.__init__ (
    self,
    pad_size = 10 )
```

This function sets the padding size for the image. Default value is 10.

### 5.1.3 Member Function Documentation

#### 5.1.3.1 `contour_len_tresh()`

```
def grass.Grass.contour_len_tresh (  
    self )
```

This function is used to set the threshold for minimum contour length for which the boundary can be drawn.

#### 5.1.3.2 `get_max_green_intensity()`

```
def grass.Grass.get_max_green_intensity (  
    self )
```

This function is used to set the maximum intensity values of the green color.  
Modify intensity value as per requirement (For different color)

#### 5.1.3.3 `get_min_green_intensity()`

```
def grass.Grass.get_min_green_intensity (  
    self )
```

This function is used to set the minimum intensity values of the green color.  
Modify intensity value as per requirement (For different color)

#### 5.1.3.4 `get_padsizes()`

```
def grass.Grass.get_padsizes (  
    self )
```

This function returns the padsizes for the image.

### 5.1.3.5 no\_dilation\_steps()

```
def grass.Grass.no_dilation_steps (
    self )
```

This function is used to set the minimum intensity values of the green color.  
Modify steps as per required no. of iterations  
Modify kernel\_size with which dilation needs to be done

### 5.1.3.6 no\_erosion\_steps()

```
def grass.Grass.no_erosion_steps (
    self )
```

This function is used to set the minimum intensity values of the green color.  
Modify steps as per required no. of iterations  
Modify kernel\_size with which dilation needs to be done

## 5.1.4 Member Data Documentation

### 5.1.4.1 pad\_size

```
grass.Grass.pad_size
```

The documentation for this class was generated from the following file:

- [grass.py](#)



## Chapter 6

# File Documentation

### 6.1 detection.py File Reference

#### Namespaces

- namespace [boundary\\_detection](#)

#### Variables

- [boundary\\_detection.parser](#) = argparse.ArgumentParser(description='Detect boundaries of the grass in the given image')
- [boundary\\_detection.type](#)
- [boundary\\_detection.int](#)
- [boundary\\_detection.required](#)
- [boundary\\_detection.help](#)
- [boundary\\_detection.str](#)
- [boundary\\_detection.True](#)
- [boundary\\_detection.False](#)
- [boundary\\_detection.args](#) = parser.parse\_args()
- [boundary\\_detection.Photos](#) = args.file\_dir
- list [boundary\\_detection.lists](#) = []
- [boundary\\_detection.image\\_array](#) = cv2.imread(Photos + file\_name)
- [boundary\\_detection.bound\\_detect](#) = Grass(10)
- [boundary\\_detection.hsv](#) = cv2.cvtColor(image\_array, cv2.COLOR\_BGR2HSV)
- [boundary\\_detection.pad\\_size](#) = bound\_detect.get\_padsizes()
- list [boundary\\_detection.BLACK](#) = [0, 0, 0]
- [boundary\\_detection.image\\_color\\_pad](#)
- [boundary\\_detection.hue](#) = hsv[:, :, 0]
- tuple [boundary\\_detection.hue\\_binary](#)
- tuple [boundary\\_detection.hue\\_numbers](#) = hue\_binary.astype(np.uint8)
- tuple [boundary\\_detection.constant](#) = hue\_numbers \* 255
- [boundary\\_detection.kernel\\_small](#) = bound\_detect.no\_dilation\_steps()[0]
- [boundary\\_detection.kernel\\_big](#) = bound\_detect.no\_erosion\_steps()[0]

- `boundary_detection.img_dilation` = `cv2.dilate(constant, kernel_small, iterations=bound_detect.no_dilation_↵ steps()[1])`
- `boundary_detection.img_erosion` = `cv2.erode(img_dilation, kernel_big, iterations=bound_detect.no_erosion_↵ steps()[1])`
- `boundary_detection.ret`
- `boundary_detection.thresh`
- `boundary_detection.contours`
- `boundary_detection.hierarchy`
- `boundary_detection.draw_img` = `image_color_pad.copy()`
- `int boundary_detection.cnt_no` = 0
- `boundary_detection.res` = `cv2.drawContours(draw_img, contours, i, (0, 0, 255), 50)`
- `boundary_detection.cmap`

## 6.2 grass.py File Reference

### Classes

- class `grass.Grass`

### Namespaces

- namespace `grass`



# Index

- [\\_\\_init\\_\\_](#)
    - [grass.Grass](#), [15](#)
- [args](#)
  - [boundary\\_detection](#), [8](#)
- [BLACK](#)
  - [boundary\\_detection](#), [8](#)
- [bound\\_detect](#)
  - [boundary\\_detection](#), [8](#)
- [boundary\\_detection](#), [7](#)
  - [args](#), [8](#)
  - [BLACK](#), [8](#)
  - [bound\\_detect](#), [8](#)
  - [cmap](#), [8](#)
  - [cnt\\_no](#), [8](#)
  - [constant](#), [9](#)
  - [contours](#), [9](#)
  - [draw\\_img](#), [9](#)
  - [False](#), [9](#)
  - [help](#), [9](#)
  - [hierarchy](#), [9](#)
  - [hsv](#), [9](#)
  - [hue](#), [10](#)
  - [hue\\_binary](#), [10](#)
  - [hue\\_numbers](#), [10](#)
  - [image\\_array](#), [10](#)
  - [image\\_color\\_pad](#), [10](#)
  - [img\\_dilation](#), [10](#)
  - [img\\_erosion](#), [11](#)
  - [int](#), [11](#)
  - [kernel\\_big](#), [11](#)
  - [kernel\\_small](#), [11](#)
  - [lists](#), [11](#)
  - [pad\\_size](#), [11](#)
  - [parser](#), [11](#)
  - [Photos](#), [12](#)
  - [required](#), [12](#)
  - [res](#), [12](#)
  - [ret](#), [12](#)
  - [str](#), [12](#)
  - [thresh](#), [12](#)
  - [True](#), [12](#)
  - [type](#), [13](#)
- [boundary\\_detection.py](#), [19](#)
- [cmap](#)
  - [boundary\\_detection](#), [8](#)
- [cnt\\_no](#)
  - [boundary\\_detection](#), [8](#)
- [constant](#)
  - [boundary\\_detection](#), [9](#)
- [contour\\_len\\_tresh](#)
  - [grass.Grass](#), [16](#)
- [contours](#)
  - [boundary\\_detection](#), [9](#)
- [draw\\_img](#)
  - [boundary\\_detection](#), [9](#)
- [False](#)
  - [boundary\\_detection](#), [9](#)
- [get\\_max\\_green\\_intensity](#)
  - [grass.Grass](#), [16](#)
- [get\\_min\\_green\\_intensity](#)
  - [grass.Grass](#), [16](#)
- [get\\_padsize](#)
  - [grass.Grass](#), [16](#)
- [grass](#), [13](#)
- [grass.Grass](#), [15](#)
  - [\\_\\_init\\_\\_](#), [15](#)
  - [contour\\_len\\_tresh](#), [16](#)
  - [get\\_max\\_green\\_intensity](#), [16](#)
  - [get\\_min\\_green\\_intensity](#), [16](#)
  - [get\\_padsize](#), [16](#)
  - [no\\_dilation\\_steps](#), [16](#)
  - [no\\_erosion\\_steps](#), [17](#)
  - [pad\\_size](#), [17](#)
- [grass.py](#), [20](#)
- [help](#)
  - [boundary\\_detection](#), [9](#)
- [hierarchy](#)
  - [boundary\\_detection](#), [9](#)
- [hsv](#)
  - [boundary\\_detection](#), [9](#)
- [hue](#)
  - [boundary\\_detection](#), [10](#)
- [hue\\_binary](#)
  - [boundary\\_detection](#), [10](#)
- [hue\\_numbers](#)

- boundary\_detection, [10](#)
- image\_array
  - boundary\_detection, [10](#)
- image\_color\_pad
  - boundary\_detection, [10](#)
- img\_dilation
  - boundary\_detection, [10](#)
- img\_erosion
  - boundary\_detection, [11](#)
- int
  - boundary\_detection, [11](#)
- kernel\_big
  - boundary\_detection, [11](#)
- kernel\_small
  - boundary\_detection, [11](#)
- lists
  - boundary\_detection, [11](#)
- no\_dilation\_steps
  - grass.Grass, [16](#)
- no\_erosion\_steps
  - grass.Grass, [17](#)
- pad\_size
  - boundary\_detection, [11](#)
  - grass.Grass, [17](#)
- parser
  - boundary\_detection, [11](#)
- Photos
  - boundary\_detection, [12](#)
- required
  - boundary\_detection, [12](#)
- res
  - boundary\_detection, [12](#)
- ret
  - boundary\_detection, [12](#)
- str
  - boundary\_detection, [12](#)
- thresh
  - boundary\_detection, [12](#)
- True
  - boundary\_detection, [12](#)
- type
  - boundary\_detection, [13](#)