Ohm - Desktop Version

1.0

Generated by Doxygen 1.8.12

Contents

1 Module Index **1**

1.1 Modules 1

2 Namespace Index **3**

2.1 Packages 3

3 Hierarchical Index **5**

3.1 Class Hierarchy 5

4 Class Index **7**

4.1 Class List 7

5 Module Documentation **9**

5.1 UserInterface 9

5.1.1 Detailed Description 9

5.2 ColourMapping 10

5.2.1 Detailed Description 10

5.3 ValueCalculator 11

5.3.1 Detailed Description 11

5.4 BandLocation 12

5.4.1 Detailed Description 12

5.5 CameraInput 13

5.5.1 Detailed Description 13

5.6 ImageInput 14

5.6.1 Detailed Description 14

6 Namespace Documentation **15**

6.1 Package imageprocessing . . . 15

6.1.1 Detailed Description . . . 15

6.2 Package input . . . 15

6.2.1 Detailed Description . . . 15

6.3 Package UserInterface . . . 15

6.3.1 Detailed Description . . . 15

6.4 Package valueidentification . . . 15

6.4.1 Detailed Description . . . 15

7 Class Documentation **17**

7.1 ohm.imageprocessing.BandReader Class Reference . . . 17

7.1.1 Detailed Description . . . 17

7.1.2 Member Function Documentation . . . 17

7.1.2.1 abmag() . . . 17

7.1.2.2 boxSample() . . . 18

7.1.2.3 componentwiseMean() . . . 18

7.1.2.4 diff() . . . 19

7.1.2.5 dist() . . . 19

7.1.2.6 findGlobalMaxima() . . . 19

7.1.2.7 findLocalMaxima() . . . 20

7.1.2.8 groupTerms() . . . 20

7.1.2.9 mag() . . . 20

7.1.2.10 onLine() . . . 21

7.1.2.11 read() . . . 21

7.1.2.12 rollingAverageFilter() . . . 21

7.1.2.13 sample() . . . 22

7.2 ohm.input.CameraInput Class Reference . . . 22

7.2.1 Detailed Description . . . 23

7.2.2 Member Function Documentation . . . 23

7.2.2.1 getImage() . . . 23

7.2.2.2 getLAB() . . . 23
7.2.2.3 getRGB() . . . 23
7.3 ohm.Helpers Class Reference . . . 24
7.3.1 Detailed Description . . . 24
7.4 ohm.input.ImageInput Class Reference . . . 24
7.4.1 Detailed Description . . . 24
7.4.2 Constructor & Destructor Documentation . . . 24
7.4.2.1 ImageInput() [1/2] . . . 24
7.4.2.2 ImageInput() [2/2] . . . 24
7.4.3 Member Function Documentation . . . 25
7.4.3.1 getImage() . . . 25
7.4.3.2 getLAB() . . . 25
7.4.3.3 getRGB() . . . 25
7.5 ohm.input.Input Interface Reference . . . 26
7.5.1 Detailed Description . . . 26
7.5.2 Member Function Documentation . . . 26
7.5.2.1 getImage() . . . 26
7.5.2.2 getLAB() . . . 26
7.5.2.3 getRGB() . . . 27
7.6 ohm.userinterface.LiveImageView Class Reference . . . 27
7.6.1 Detailed Description . . . 27
7.7 ohm.Ohm Class Reference . . . 28
7.8 ohm.userinterface.OhmViewController Class Reference . . . 28
7.8.1 Detailed Description . . . 28
7.8.2 Member Function Documentation . . . 29
7.8.2.1 initialize() . . . 29
7.9 ohm.userinterface.ResistorAxisPickerView Class Reference . . . 29
7.9.1 Detailed Description . . . 29
7.9.2 Member Function Documentation . . . 29
7.9.2.1 setImage() . . . 29

7.9.2.2 setListener() . . . 30

7.10 ohm.valueidentification.ResistorColour Enum Reference . . . 30

7.10.1 Detailed Description . . . 31

7.10.2 Constructor & Destructor Documentation . . . 31

7.10.2.1 ResistorColour() . . . 31

7.10.3 Member Function Documentation . . . 31

7.10.3.1 [static initializer]() . . . 31

7.10.3.2 fit() [1/2] . . . 31

7.10.3.3 fit() [2/2] . . . 32

7.10.3.4 rgb2lab() . . . 32

7.11 ohm.valueidentification.ValueCalculator Class Reference . . . 32

7.11.1 Detailed Description . . . 33

7.11.2 Constructor & Destructor Documentation . . . 33

7.11.2.1 ValueCalculator() . . . 33

7.11.3 Member Function Documentation . . . 33

7.11.3.1 getValue() . . . 33

Index **35**

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

UserInterface 9
ColourMapping 10
ValueCalculator 11
BandLocation 12
CameraInput 13
ImageInput 14

Chapter 2

Namespace Index

2.1 Packages

Here are the packages with brief descriptions (if available):

imageprocessing 15
input 15
UserInterface 15
valueidentification 15

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

ohm.imageprocessing.BandReader 17
ohm.Helpers 24
ohm.input.Input 26
    ohm.input.CameraInput 22
    ohm.input.ImageInput 24
ohm.valueidentification.ResistorColour 30
ohm.valueidentification.ValueCalculator 32
Application
    ohm.Ohm 28
EventHandler
    ohm.userinterface.ResistorAxisPickerView 29
ImageView
    ohm.userinterface.LiveImageView 27
    ohm.userinterface.ResistorAxisPickerView 29
Initializable
    ohm.userinterface.OhmViewController 28

Chapter 4

Class Index

4.1 Class List

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

ohm.imageprocessing.BandReader
: Module used to analyze the line of pixels selected by the user through the UI. It uses high values of the differential of the RGB colours to detect edges of bands . . . 17

ohm.input.CameraInput
: Class used to receive input from the camera on Desktop . . . 22

ohm.Helpers . . . 24

ohm.input.ImageInput
: A source of input data, uses static images . . . 24

ohm.input.Input . . . 26

ohm.userinterface.LiveImageView
: The LiveImageView is an ImageView that periodically calls a Renderer to produce a new image that the ImageView then displays. This is useful to create images from real time updating images/data . . . 27

ohm.Ohm . . . 28

ohm.userinterface.OhmViewController
: Coordinates the algorithms present in the model with user input through the view . . . 28

ohm.userinterface.ResistorAxisPickerView
: This class displays an opencv Mat as an image and allows the user to pick two endpoints of a line. Once two valid points, a listener is called . . . 29

ohm.valueidentification.ResistorColour
: Enum containing all of the possible colours that a resistor can take on. Also features member functions used to map the colours of bands to values used in the calculation process . . . 30

ohm.valueidentification.ValueCalculator
: Object used to calculate the resistance of the resistor based on the mapped colours . . . 32

Chapter 5

Module Documentation

5.1 UserInterface

Classes

- class ohm.userinterface.LiveImageView

  The LiveImageView is an ImageView that periodically calls a Renderer to produce a new image that the ImageView then displays. This is useful to create images from real time updating images/data.
- class ohm.userinterface.OhmViewController

  Coordinates the algorithms present in the model with user input through the view.
- class ohm.userinterface.ResistorAxisPickerView

  This class displays an opencv Mat as an image and allows the user to pick two endpoints of a line. Once two valid points, a listener is called.

5.1.1 Detailed Description

5.2 ColourMapping

Classes

- enum ohm.valueidentification.ResistorColour

  Enum containing all of the possible colours that a resistor can take on. Also features member functions used to map the colours of bands to values used in the calculation process.

5.2.1 Detailed Description

Author

Jonathan Brown

5.3 ValueCalculator

Classes

- class ohm.valueidentification.ValueCalculator

  Object used to calculate the resistance of the resistor based on the mapped colours.

5.3.1 Detailed Description

Author

Jonathan Brown

5.4 BandLocation

Classes

- class ohm.imageprocessing.BandReader

  Module used to analyze the line of pixels selected by the user through the UI. It uses high values of the differential of the RGB colours to detect edges of bands.

5.4.1 Detailed Description

Module used to analyze the line of pixels selected by the user through the UI. It uses high values of the differential of the RGB colours to detect edges of bands.

5.5 CameraInput

Classes

- class ohm.input.CameraInput

  Class used to receive input from the camera on Desktop.

5.5.1 Detailed Description

Author

Graeme Crawley

5.6 ImageInput

Classes

- class ohm.input.ImageInput

  A source of input data, uses static images.

5.6.1 Detailed Description

Chapter 6

Namespace Documentation

6.1 Package imageprocessing

6.1.1 Detailed Description

Contains the Band Location and Resistor Body Identification modules.

6.2 Package input

6.2.1 Detailed Description

This package contains the components of the Camera and ImageFile modules. They hide the applications access to the hardware (camera or file system).

6.3 Package UserInterface

6.3.1 Detailed Description

Contains the views and the controller of the application. This is a Behavioural Module who's secret is the UI components of the application.

6.4 Package valueidentification

6.4.1 Detailed Description

Contains the Colour Mapping and Value Identification Modules

Chapter 7

Class Documentation

7.1 ohm.imageprocessing.BandReader Class Reference

Module used to analyze the line of pixels selected by the user through the UI. It uses high values of the differential of the RGB colours to detect edges of bands.

Static Public Member Functions

- static List< Point > read (Mat frame, Point p1, Point p2)
- static double [ ][ ] boxSample (Mat mat, Point start, Point end, int nSamples, double boxWidth)
- static double [ ][ ] rollingAverageFilter (double[ ][ ] sample, int windowRadius)
- static double [ ] componentwiseMean (double[ ][ ] samples)
- static double [ ][ ] sample (Mat mat, Point start, Point end, int nSamples)
- static double [ ][ ] diff (double[ ][ ] y)
- static double abmag (double[ ] vect)
- static double mag (double[ ] vect)
- static double dist (Point p1, Point p2)
- static double [ ] groupTerms (double[ ] terms, int binSize)
- static List< Pair< Integer, Double > > findLocalMaxima (double[ ] values)
- static List< Pair< Integer, Double > > findGlobalMaxima (double[ ] values, int nMaxima)
- static Point onLine (Point start, Point end, double fraction)

7.1.1 Detailed Description

Module used to analyze the line of pixels selected by the user through the UI. It uses high values of the differential of the RGB colours to detect edges of bands.

Author

Ryan Marks & Jonathan Brown

7.1.2 Member Function Documentation

7.1.2.1 abmag()

```
static double ohm.imageprocessing.BandReader.abmag (
            double [ ] vect )  [static]
```

Calculate the magnitude of a given vector

Parameters

vect	input vector

Returns

The magnitude of the input vector

7.1.2.2 boxSample()

```
static double [][] ohm.imageprocessing.BandReader.boxSample (
            Mat mat,
            Point start,
            Point end,
            int nSamples,
            double boxWidth )  [static]
```

This method takes something Ryan made up called a box sample. At nSamples points along the line from start to end, a line of samples are take along a line of length boxWidth, normal to and centred on the sampling line

Parameters

mat	The matrix being sampled
start	The start of the sampling line
end	The end of the sampling line
nSamples	The number or linear samples to take along the sampling line
boxWidth	the length of the normal sampling line.

Returns

An array containing the average rgb values (represented as double[]) in the sampling window.

7.1.2.3 componentwiseMean()

```
static double [] ohm.imageprocessing.BandReader.componentwiseMean (
            double samples[][] )  [static]
```

Calculate the mean of an array of vectors.

Parameters

samples	Array of RGB vectors represented as double[]

Returns

The average of all vectors within the input array (represented as a double[])

7.1.2.4 diff()

```
static double [][] ohm.imageprocessing.BandReader.diff (
            double y[][] )  [static]
```

Take the discrete derivative of a series of vectors.

Parameters

y	A series of vectors

Returns

y', the derivative of the input represented as a array of vectors (double[]).

7.1.2.5 dist()

```
static double ohm.imageprocessing.BandReader.dist (
            Point p1,
            Point p2 )  [static]
```

Calcualte the distance between two points.

Parameters

p1	Point 1.
p2	Point 2.

Returns

Distance between p1 and p2.

7.1.2.6 findGlobalMaxima()

```
static List<Pair<Integer,Double> > ohm.imageprocessing.BandReader.findGlobalMaxima (
            double [] values,
            int nMaxima )  [static]
```

Get the n largest values in a sample/

Parameters

values	the input signal
nMaxima	The number of maxima to find

Returns

A list of (index, value) Pairs representing the global maxima in the sample

7.1.2.7 findLocalMaxima()

```
static List<Pair<Integer,Double> > ohm.imageprocessing.BandReader.findLocalMaxima (
            double [] values )  [static]
```

Find all the local maxima in an array of doubles

Parameters

values	the input signal

Returns

A list of (index, value) Pairs representing the local maxima

7.1.2.8 groupTerms()

```
static double [] ohm.imageprocessing.BandReader.groupTerms (
            double [] terms,
            int binSize )  [static]
```

Take a large array of samples and collect the samples into bins

Parameters

terms	The array of samples
binSize	The size of the bins

Returns

The binned samples

7.1.2.9 mag()

```
static double ohm.imageprocessing.BandReader.mag (
            double [] vect )  [static]
```

Calculate the magnitude of a given vector

Parameters

vect	input vector

Returns

The magnitude of the input vector

7.1.2.10 onLine()

```
static Point ohm.imageprocessing.BandReader.onLine (
            Point start,
            Point end,
            double fraction )  [static]
```

Get a point that is some fraction between two others.

Parameters

start	The starting point of the line
end	The endpoint of the line
fraction	The fraction of the way between start and end

Returns

A point that is fraction between start and end

7.1.2.11 read()

```
static List<Point> ohm.imageprocessing.BandReader.read (
            Mat frame,
            Point p1,
            Point p2 )  [static]
```

Parameters

frame	Image to Sample.
p1	Starting point of the sampling line.
p2	Ending point of the sampling line.

Returns

List of points along the line that are likely band edges.

7.1.2.12 rollingAverageFilter()

```
static double [][] ohm.imageprocessing.BandReader.rollingAverageFilter (
            double sample[][],
            int windowRadius )  [static]
```

This method is used to apply a rolling average filter to vector data.

Parameters

sample	An array of (arrays of doubles representing an individual vector sample)
windowRadius	The number of samples on either side of a given sample to incorporate into the average

Returns

An array containing the average RGB value sampled along the line.

7.1.2.13 sample()

```
static double [][] ohm.imageprocessing.BandReader.sample (
            Mat mat,
            Point start,
            Point end,
            int nSamples )  [static]
```

Take a linear sampling from a matrix at a given number of points along a line

Parameters

mat	The matrix to be sampled
start	The start of the sampling line
end	The end of the sampling line
nSamples	The number of samples to take along the line.

Returns

The samples taken

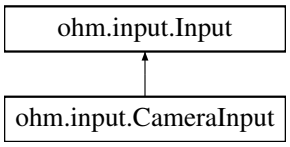
The documentation for this class was generated from the following file:

- java/ohm/imageprocessing/BandReader.java

7.2 ohm.input.CameraInput Class Reference

Class used to receive input from the camera on Desktop.

Inheritance diagram for ohm.input.CameraInput:



Public Member Functions

- Image getImage ()
- Mat getRGB ()
- Mat getLAB ()

7.2.1 Detailed Description

Class used to receive input from the camera on Desktop.

7.2.2 Member Function Documentation

7.2.2.1 getImage()

```
Image ohm.input.CameraInput.getImage ( )
```

Used to retrieve a image representation of the input (used by JavaFX).

Returns

Image representation of input.

Implements ohm.input.Input.

7.2.2.2 getLAB()

```
Mat ohm.input.CameraInput.getLAB ( )
```

Used to retrieve a rgb matrix representation of the input (used by Opencv libraries).

Returns

Matrix representation of input.

Implements ohm.input.Input.

7.2.2.3 getRGB()

```
Mat ohm.input.CameraInput.getRGB ( )
```

Used to retrieve a rgb matrix representation of the input (used by Opencv libraries).

Returns

Matrix representation of input.

Implements ohm.input.Input.

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

- java/ohm/input/CameraInput.java

7.3 ohm.Helpers Class Reference

Static Public Member Functions

- static Image **matToImage** (Mat mat)

7.3.1 Detailed Description

Module featuring static methods used by multiple other modules in order to perform common computations. Will be eliminated in final release.

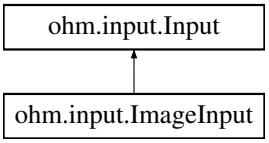
The documentation for this class was generated from the following file:

- java/ohm/Helpers.java

7.4 ohm.input.ImageInput Class Reference

A source of input data, uses static images.

Inheritance diagram for ohm.input.ImageInput:



Public Member Functions

- ImageInput ()
- ImageInput (String name)
- Mat getRGB ()
- Image getImage ()
- Mat getLAB ()

7.4.1 Detailed Description

A source of input data, uses static images.

Author

Jonathan Brown

7.4.2 Constructor & Destructor Documentation

7.4.2.1 ImageInput() [1/2]

```
ohm.input.ImageInput.ImageInput ( )
```

Default constructor creates an instance using a default image.

7.4.2.2 ImageInput() [2/2]

```
ohm.input.ImageInput.ImageInput (
            String name )
```

Constructor featuring a name parameter.

Parameters

name	The name of the input image. Do not include an extension.

7.4.3 Member Function Documentation

7.4.3.1 getImage()

```
Image ohm.input.ImageInput.getImage ( )
```

Used to retrieve a image representation of the input (used by JavaFX).

Returns

Image representation of input.

Implements ohm.input.Input.

7.4.3.2 getLAB()

```
Mat ohm.input.ImageInput.getLAB ( )
```

Used to retrieve a rgb matrix representation of the input (used by Opencv libraries).

Returns

Matrix representation of input.

Implements ohm.input.Input.

7.4.3.3 getRGB()

```
Mat ohm.input.ImageInput.getRGB ( )
```

Method returns a OpenCV Matrix of the loaded image.

Returns

OpenCV Matrix representation of the image.

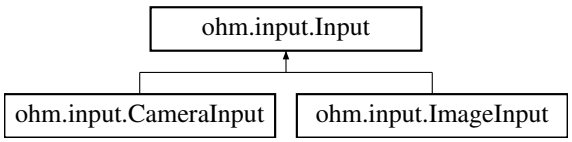
Implements ohm.input.Input.

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

- java/ohm/input/ImageInput.java

7.5 ohm.input.Input Interface Reference

Inheritance diagram for ohm.input.Input:



Public Member Functions

- Mat getRGB ()
- Image getImage ()
- Mat getLAB ()

7.5.1 Detailed Description

Author

Jonathan Brown

7.5.2 Member Function Documentation

7.5.2.1 getImage()

```
Image ohm.input.Input.getImage ( )
```

Used to retrieve a image representation of the input (used by JavaFX).

Returns

Image representation of input.

Implemented in ohm.input.CameraInput, and ohm.input.ImageInput.

7.5.2.2 getLAB()

```
Mat ohm.input.Input.getLAB ( )
```

Used to retrieve a rgb matrix representation of the input (used by Opencv libraries).

Returns

Matrix representation of input.

Implemented in ohm.input.CameraInput, and ohm.input.ImageInput.

7.5.2.3 getRGB()

```
Mat ohm.input.Input.getRGB ( )
```

Used to retrieve a rgb matrix representation of the input (used by Opencv libraries).

Returns

Matrix representation of input.

Implemented in ohm.input.CameraInput, and ohm.input.ImageInput.

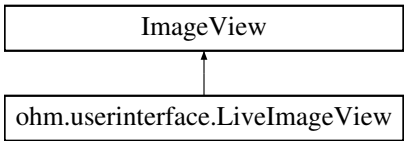
The documentation for this interface was generated from the following file:

- java/ohm/input/Input.java

7.6 ohm.userinterface.LiveImageView Class Reference

The LiveImageView is an ImageView that periodically calls a Renderer to produce a new image that the ImageView then displays. This is useful to create images from real time updating images/data.

Inheritance diagram for ohm.userinterface.LiveImageView:



Classes

- interface **Renderer**

Public Member Functions

- void **setFrameRate** (int frameRate)
- void **setRenderer** (Renderer newRenderer)

7.6.1 Detailed Description

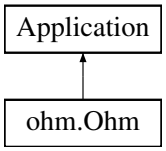
The LiveImageView is an ImageView that periodically calls a Renderer to produce a new image that the ImageView then displays. This is useful to create images from real time updating images/data.

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

- java/ohm/userinterface/LiveImageView.java

7.7 ohm.Ohm Class Reference

Inheritance diagram for ohm.Ohm:



Public Member Functions

- void **start** (Stage primaryStage) throws Exception

Static Public Member Functions

- static void **main** (String[ ] args) throws Exception

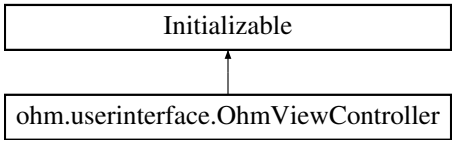
The documentation for this class was generated from the following file:

- java/ohm/Ohm.java

7.8 ohm.userinterface.OhmViewController Class Reference

Coordinates the algorithms present in the model with user input through the view.

Inheritance diagram for ohm.userinterface.OhmViewController:



Public Member Functions

- void initialize (URL url, ResourceBundle rb)
- void **buttonClicked** (ActionEvent actionEvent)

Public Attributes

- Button **startCameraButton**

7.8.1 Detailed Description

Coordinates the algorithms present in the model with user input through the view.

7.8.2 Member Function Documentation

7.8.2.1 initialize()

```
void ohm.userinterface.OhmViewController.initialize (
            URL url,
            ResourceBundle rb )
```

Method used to "glue" together the front and back end of the application.

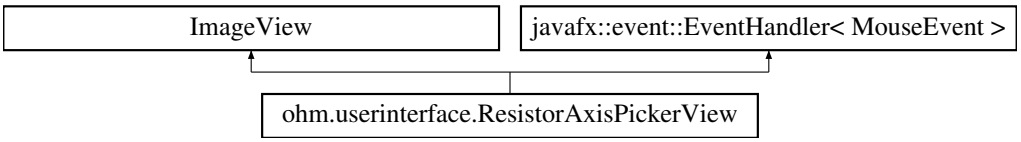
The documentation for this class was generated from the following file:

- java/ohm/userinterface/OhmViewController.java

7.9 ohm.userinterface.ResistorAxisPickerView Class Reference

This class displays an opencv Mat as an image and allows the user to pick two endpoints of a line. Once two valid points, a listener is called.

Inheritance diagram for ohm.userinterface.ResistorAxisPickerView:



Classes

- interface **Listener**

Public Member Functions

- void **handle** (MouseEvent event)
- void setImage (Mat newImage)
- void setListener (Listener listener)

Static Public Member Functions

- static double **dist** (Point p1, Point p2)

7.9.1 Detailed Description

This class displays an opencv Mat as an image and allows the user to pick two endpoints of a line. Once two valid points, a listener is called.

7.9.2 Member Function Documentation

7.9.2.1 setImage()

```
void ohm.userinterface.ResistorAxisPickerView.setImage (
            Mat newImage )
```

Set the matrix to display

Parameters

newImage	Image to be set as the image to be displayed in the UI

7.9.2.2 setListener()

```
void ohm.userinterface.ResistorAxisPickerView.setListener (
            Listener listener )
```

Assign a line listener to be updated when new points are selected

Parameters

listener	Object that triggers responds to a mouse click inside the view.

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

- java/ohm/userinterface/ResistorAxisPickerView.java

7.10 ohm.valueidentification.ResistorColour Enum Reference

Enum containing all of the possible colours that a resistor can take on. Also features member functions used to map the colours of bands to values used in the calculation process.

Public Member Functions

- ResistorColour (int v)

Static Public Member Functions

- [static initializer]
- static void rgb2lab (float R, float G, float B, float[ ] lab)
- static int **fit** (float r, float g, float b)
- static int fit (int r, int g, int b, int colorSpace)
- static int fit (double r, double g, double b, int colorSpace)

Public Attributes

- **BLACK** =(0)
- **BROWN** =(1)
- **RED** =(2)
- **ORANGE** =(3)
- **YELLOW** =(4)
- **GREEN** =(5)
- **BLUE** =(6)
- **VIOLET** =(7)
- **GREY** =(8)
- **WHITE** =(9)
- **GOLD** =(11)
- **BASE** =(10)
- int **value**

Static Public Attributes

- static KNearest **KNN**

7.10.1 Detailed Description

Enum containing all of the possible colours that a resistor can take on. Also features member functions used to map the colours of bands to values used in the calculation process.

7.10.2 Constructor & Destructor Documentation

7.10.2.1 ResistorColour()

```
ohm.valueidentification.ResistorColour.ResistorColour (
            int v )
```

Parameters

v	The number represented by the colour in the calculation of the resistor's ohmage.

-

7.10.3 Member Function Documentation

7.10.3.1 [static initializer]()

```
ohm.valueidentification.ResistorColour.[static initializer] ( )  [static]
```

Creates and trains the KNN classifier. Based on the training set stored in train.csv.

7.10.3.2 fit() [1/2]

```
static int ohm.valueidentification.ResistorColour.fit (
            int r,
            int g,
            int b,
            int colorSpace )  [static]
```

Function takes in a sampled colour from the images and attempts to fitOld it to the closest known colour a resistor can possess.

Parameters

r	The red colour value of the colour to be fitOld.
g	The green colour value of the colour to be fitOld.
b	The blue colour value of the colour to be fitOld.

Returns

The known colour that best represents the sampled colour.

7.10.3.3 fit() [2/2]

```
static int ohm.valueidentification.ResistorColour.fit (
            double r,
            double g,
            double b,
            int colorSpace )  [static]
```

Function takes in a sampled colour from the images and attempts to fitOld it to the closest known colour a resistor can possess.

Parameters

r	The red colour value of the colour to be fitOld.
g	The green colour value of the colour to be fitOld.
b	The blue colour value of the colour to be fitOld.

Returns

The known colour that best represents the sampled colour.

7.10.3.4 rgb2lab()

```
static void ohm.valueidentification.ResistorColour.rgb2lab (
            float R,
            float G,
            float B,
            float [] lab )  [static]
```

Method to convert a pixel from RGB space to lab space. Credit to: //http://www.brucelindbloom.com

Parameters

R	Red value from 0f to 255f
G	Green value from 0f to 255f
B	Blue value from 0f to 255f
lab	Size three float[] containing the lab representation of the pixel.

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

- java/ohm/valueidentification/ResistorColour.java

7.11 ohm.valueidentification.ValueCalculator Class Reference

Object used to calculate the resistance of the resistor based on the mapped colours.

Public Member Functions

- ValueCalculator (Integer a, Integer b, Integer c, Integer d)
- String getValue ()

7.11.1 Detailed Description

Object used to calculate the resistance of the resistor based on the mapped colours.

Author

Jonathan Brown

7.11.2 Constructor & Destructor Documentation

7.11.2.1 ValueCalculator()

```
ohm.valueidentification.ValueCalculator.ValueCalculator (
            Integer a,
            Integer b,
            Integer c,
            Integer d )
```

Parameters

a	Leftmost band value
b	2nd leftmost band value
c	3rd leftmost band value
d	Rightmost band value

7.11.3 Member Function Documentation

7.11.3.1 getValue()

```
String ohm.valueidentification.ValueCalculator.getValue ( )
```

Method to obtain the ohmage of the resistor calculated by the ValueCalculator.

Returns

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

- java/ohm/valueidentification/ValueCalculator.java

Index

[static initializer]
    ohm::valueidentification::ResistorColour, 31

abmag
    ohm::imageprocessing::BandReader, 17

BandLocation, 12
boxSample
    ohm::imageprocessing::BandReader, 18

CameraInput, 13
ColourMapping, 10
componentwiseMean
    ohm::imageprocessing::BandReader, 18

diff
    ohm::imageprocessing::BandReader, 18
dist
    ohm::imageprocessing::BandReader, 19

findGlobalMaxima
    ohm::imageprocessing::BandReader, 19
findLocalMaxima
    ohm::imageprocessing::BandReader, 19
fit
    ohm::valueidentification::ResistorColour, 31, 32

getImage
    ohm::input::CameraInput, 23
    ohm::input::ImageInput, 25
    ohm::input::Input, 26
getLAB
    ohm::input::CameraInput, 23
    ohm::input::ImageInput, 25
    ohm::input::Input, 26
getRGB
    ohm::input::CameraInput, 23
    ohm::input::ImageInput, 25
    ohm::input::Input, 26
getValue
    ohm::valueidentification::ValueCalculator, 33
groupTerms
    ohm::imageprocessing::BandReader, 20

ImageInput, 14
    ohm::input::ImageInput, 24
imageprocessing, 15
initialize
    ohm::userinterface::OhmViewController, 29
input, 15

mag
    ohm::imageprocessing::BandReader, 20

ohm.Helpers, 24
ohm.imageprocessing.BandReader, 17
ohm.input.CameraInput, 22
ohm.input.ImageInput, 24
ohm.input.Input, 26
ohm.Ohm, 28
ohm.userinterface.LiveImageView, 27
ohm.userinterface.OhmViewController, 28
ohm.userinterface.ResistorAxisPickerView, 29
ohm.valueidentification.ResistorColour, 30
ohm.valueidentification.ValueCalculator, 32
ohm::imageprocessing::BandReader
    abmag, 17
    boxSample, 18
    componentwiseMean, 18
    diff, 18
    dist, 19
    findGlobalMaxima, 19
    findLocalMaxima, 19
    groupTerms, 20
    mag, 20
    onLine, 20
    read, 21
    rollingAverageFilter, 21
    sample, 22
ohm::input::CameraInput
    getImage, 23
    getLAB, 23
    getRGB, 23
ohm::input::ImageInput
    getImage, 25
    getLAB, 25
    getRGB, 25
    ImageInput, 24
ohm::input::Input
    getImage, 26
    getLAB, 26
    getRGB, 26
ohm::userinterface::OhmViewController
    initialize, 29
ohm::userinterface::ResistorAxisPickerView
    setImage, 29
    setListener, 30
ohm::valueidentification::ResistorColour
    [static initializer], 31
    fit, 31, 32
    ResistorColour, 31

rgb2lab, 32
ohm::valueidentification::ValueCalculator
getValue, 33
ValueCalculator, 33
onLine
ohm::imageprocessing::BandReader, 20

read
ohm::imageprocessing::BandReader, 21
ResistorColour
ohm::valueidentification::ResistorColour, 31
rgb2lab
ohm::valueidentification::ResistorColour, 32
rollingAverageFilter
ohm::imageprocessing::BandReader, 21

sample
ohm::imageprocessing::BandReader, 22
setImage
ohm::userinterface::ResistorAxisPickerView, 29
setListener
ohm::userinterface::ResistorAxisPickerView, 30

UserInterface, 9, 15

ValueCalculator, 11
ohm::valueidentification::ValueCalculator, 33
valueidentification, 15