

Head Circumference Embedded

Generated by Doxygen 1.9.7

1 machine learnig source code	1
2 Source Files	3
3 Namespace Index	5
3.1 Namespace List	5
4 Class Index	7
4.1 Class List	7
5 File Index	9
5.1 File List	9
6 Namespace Documentation	11
6.1 ActiveGeomtericShape Namespace Reference	11
6.1.1 Detailed Description	11
6.2 MachineLearning Namespace Reference	11
7 Class Documentation	13
7.1 ActiveGeomtericShape::Ellipse Class Reference	13
7.1.1 Detailed Description	13
7.1.2 Member Function Documentation	13
7.1.2.1 calculate_force_field()	13
7.2 ActiveGeomtericShape::Ellipse::Image Struct Reference	14
7.2.1 Member Data Documentation	14
7.2.1.1 height	14
7.2.1.2 width	14
7.2.1.3 x	14
7.3 MachineLearning::Image Class Reference	14
7.3.1 Constructor & Destructor Documentation	15
7.3.1.1 Image()	15
7.3.1.2 ~Image()	15
7.3.2 Member Function Documentation	15
7.3.2.1 load_data()	15
7.3.2.2 safe_append()	15
7.3.2.3 set_test_status()	15
7.4 MachineLearning::Logger Class Reference	15
7.4.1 Member Function Documentation	16
7.4.1.1 display_error_message()	16
7.4.1.2 display_info_message()	16
7.4.1.3 display_test_message()	16
7.4.1.4 set_levels()	16
7.5 MachineLearning::Unet Class Reference	16
7.5.1 Constructor & Destructor Documentation	16

7.5.1.1 Unet()	16
7.5.1.2 ~Unet()	16
8 File Documentation	17
8.1 src/active_geometric_shape/ellipse.cpp File Reference	17
8.2 src/active_geometric_shape/ellipse.h File Reference	17
8.3 ellipse.h	18
8.4 src/head_circumference.cpp File Reference	18
8.4.1 Function Documentation	18
8.4.1.1 main()	18
8.5 src/machine_learning/image.cpp File Reference	18
8.6 src/machine_learning/image.h File Reference	19
8.7 image.h	19
8.8 src/machine_learning/logger.cpp File Reference	19
8.9 src/machine_learning/logger.h File Reference	20
8.10 logger.h	20
8.11 src/machine_learning/README.md File Reference	20
8.12 src/README.md File Reference	20
8.13 src/machine_learning/unet.cpp File Reference	20
8.14 src/machine_learning/unet.h File Reference	21
8.15 unet.h	21
Index	23

Chapter 1

machine learnig source code

classes here are based on [dlib](#)

Chapter 2

Source Files

These are the source files used by [Makefile](#).

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

ActiveGeometricShape	
Classes to fit predefined shapes to data	11
MachineLearning	11

Chapter 4

Class Index

4.1 Class List

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

ActiveGeomtericShape::Ellipse	
To find and fit an ellipse in 2D image data	13
ActiveGeomtericShape::Ellipse::Image	14
MachineLearning::Image	14
MachineLearning::Logger	15
MachineLearning::Unet	16

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

src/ head_circumference.cpp	18
src/active_geometric_shape/ ellipse.cpp	17
src/active_geometric_shape/ ellipse.h	17
src/machine_learning/ image.cpp	18
src/machine_learning/ image.h	19
src/machine_learning/ logger.cpp	19
src/machine_learning/ logger.h	20
src/machine_learning/ unet.cpp	20
src/machine_learning/ unet.h	21

Chapter 6

Namespace Documentation

6.1 ActiveGeomtericShape Namespace Reference

Classes to fit predefined shapes to data.

Classes

- class [Ellipse](#)
To find and fit an ellipse in 2D image data.

6.1.1 Detailed Description

Classes to fit predefined shapes to data.

Based on the article by Wang et al. (2012)

6.2 MachineLearning Namespace Reference

Classes

- class [Image](#)
- class [Logger](#)
- class [Unet](#)

Chapter 7

Class Documentation

7.1 ActiveGeomtericShape::Ellipse Class Reference

To find and fit an ellipse in 2D image data.

```
#include <ellipse.h>
```

Classes

- struct [Image](#)

Public Member Functions

- void [calculate_force_field](#) ([Ellipse::Image](#) i)

7.1.1 Detailed Description

To find and fit an ellipse in 2D image data.

Based on the article by Wang et al. (2012)

7.1.2 Member Function Documentation

7.1.2.1 calculate_force_field()

```
void ActiveGeomtericShape::Ellipse::calculate_force_field (  
    Ellipse::Image i )
```

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

- src/active_geometric_shape/[ellipse.h](#)
- src/active_geometric_shape/[ellipse.cpp](#)

7.2 ActiveGeomtericShape::Ellipse::Image Struct Reference

```
#include <ellipse.h>
```

Public Attributes

- `size_t` [width](#)
- `size_t` [height](#)
- `std::vector< double >` [x](#)

7.2.1 Member Data Documentation

7.2.1.1 height

```
size_t ActiveGeomtericShape::Ellipse::Image::height
```

7.2.1.2 width

```
size_t ActiveGeomtericShape::Ellipse::Image::width
```

7.2.1.3 x

```
std::vector<double> ActiveGeomtericShape::Ellipse::Image::x
```

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

- [src/active_geometric_shape/ellipse.h](#)

7.3 MachineLearning::Image Class Reference

```
#include <image.h>
```

Public Member Functions

- [Image](#) (std::string ultrasound_folder)
- [~Image](#) ()
- void [load_data](#) (std::string filename)
- std::filesystem::path [safe_append](#) (std::filesystem::path original, std::string addition)
- void [set_test_status](#) (bool test_status)

7.3.1 Constructor & Destructor Documentation

7.3.1.1 Image()

```
MachineLearning::Image::Image (
    std::string ultrasound_folder )
```

7.3.1.2 ~Image()

```
MachineLearning::Image::~Image ( )
```

7.3.2 Member Function Documentation

7.3.2.1 load_data()

```
void MachineLearning::Image::load_data (
    std::string filename )
```

7.3.2.2 safe_append()

```
std::filesystem::path MachineLearning::Image::safe_append (
    std::filesystem::path original,
    std::string addition )
```

7.3.2.3 set_test_status()

```
void MachineLearning::Image::set_test_status (
    bool test_status )
```

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

- [src/machine_learning/image.h](#)
- [src/machine_learning/image.cpp](#)

7.4 MachineLearning::Logger Class Reference

```
#include <logger.h>
```

Static Public Member Functions

- static void [display_error_message](#) (std::string error)
- static void [display_info_message](#) (std::string info)
- static void [display_test_message](#) ()
- static void [set_levels](#) ()

7.4.1 Member Function Documentation

7.4.1.1 display_error_message()

```
void MachineLearning::Logger::display_error_message (
    std::string error ) [static]
```

7.4.1.2 display_info_message()

```
void MachineLearning::Logger::display_info_message (
    std::string info ) [static]
```

7.4.1.3 display_test_message()

```
void MachineLearning::Logger::display_test_message ( ) [static]
```

7.4.1.4 set_levels()

```
void MachineLearning::Logger::set_levels ( ) [static]
```

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

- [src/machine_learning/logger.h](#)
- [src/machine_learning/logger.cpp](#)

7.5 MachineLearning::Unet Class Reference

```
#include <unet.h>
```

Public Member Functions

- [Unet\(\)](#)
- [~Unet\(\)](#)

7.5.1 Constructor & Destructor Documentation

7.5.1.1 Unet()

```
MachineLearning::Unet::Unet ( )
```

7.5.1.2 ~Unet()

```
MachineLearning::Unet::~~Unet ( )
```

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

- [src/machine_learning/unet.h](#)
- [src/machine_learning/unet.cpp](#)

Chapter 8

File Documentation

8.1 src/active_geometric_shape/ellipse.cpp File Reference

```
#include "active_geometric_shape/ellipse.h"
```

Namespaces

- namespace [ActiveGeometricShape](#)
Classes to fit predefined shapes to data.

8.2 src/active_geometric_shape/ellipse.h File Reference

```
#include <vector>
```

Classes

- class [ActiveGeometricShape::Ellipse](#)
To find and fit an ellipse in 2D image data.
- struct [ActiveGeometricShape::Ellipse::Image](#)

Namespaces

- namespace [ActiveGeometricShape](#)
Classes to fit predefined shapes to data.

8.3 ellipse.h

[Go to the documentation of this file.](#)

```
00001 //Copyright (2023) Dr. David A. Magezi
00002 #ifndef DEF_AGS_ELLIPSE
00003 #define DEF_AGS_ELLIPSE
00004
00005 #include <vector>
00011 namespace ActiveGeometricShape{
00017 class Ellipse{
00018 public:
00019     struct Image{
00020         size_t width;
00021         size_t height;
00022         std::vector<double> x;
00023     };
00024     void calculate_force_field(Ellipse::Image i);
00025
00026 private:
00027     Ellipse::Image force_field_;
00028 };
00029 }
00030 }//namespace ActiveGeometricShape
00031 #endif //DEF_AGS_ELLIPSE
00032
```

8.4 src/head_circumference.cpp File Reference

```
#include <filesystem>
#include <string>
#include "active_geometric_shape/ellipse.h"
#include "machine_learning/image.h"
#include "machine_learning/logger.h"
#include "machine_learning/unet.h"
```

Functions

- `int main (int argc, char **argv)`

8.4.1 Function Documentation

8.4.1.1 main()

```
int main (
    int argc,
    char ** argv )
```

8.5 src/machine_learning/image.cpp File Reference

```
#include <sstream>
#include <dlib/image_io.h>
#include "machine_learning/image.h"
#include "machine_learning/logger.h"
```

Namespaces

- namespace [MachineLearning](#)

8.6 src/machine_learning/image.h File Reference

```
#include <filesystem>
#include <string>
#include <vector>
```

Classes

- class [MachineLearning::Image](#)

Namespaces

- namespace [MachineLearning](#)

8.7 image.h

[Go to the documentation of this file.](#)

```
00001 //Copyright (2023) Dr. David A. Magezi
00002
00003 #ifndef DEF_ML_IMAGE
00004 #define DEF_ML_IMAGE
00005
00006 #include <filesystem>
00007 #include <string>
00008 #include <vector>
00009
00010
00011 namespace MachineLearning{
00012 class Image{
00013 public:
00014     Image(std::string ultrasound_folder);
00015     ~Image();
00016
00017     void load_data(std::string filename);
00018     std::filesystem::path safe_append(std::filesystem::path original, std::string addition);
00019     void set_test_status(bool test_status);
00020
00021 private:
00022     std::filesystem::path get_mask_path();
00023
00024     bool test_status_;
00025     std::filesystem::path current_subfolder_, image_path_;
00026     inline static const std::string image_extension_ = "png";
00027     inline static const std::string mask_suffix_ = "_Annotation";
00028     inline static const std::string mask_tail_ = mask_suffix_ + "." + image_extension_;
00029     inline static const std::string test_subfolder_ = "test_set/";
00030     inline static const std::string training_subfolder_ = "training_set/";
00031     std::filesystem::path ultrasound_folder_;
00032 };
00033 }//namespace MachineLearning
00034
00035 #endif //DEF_ML_IMAGE
```

8.8 src/machine_learning/logger.cpp File Reference

```
#include <dlib/misc_api.h>
#include "machine_learning/logger.h"
```

Namespaces

- namespace [MachineLearning](#)

8.9 src/machine_learning/logger.h File Reference

```
#include <string>
#include <dlib/logger.h>
```

Classes

- class [MachineLearning::Logger](#)

Namespaces

- namespace [MachineLearning](#)

8.10 logger.h

[Go to the documentation of this file.](#)

```
00001 //Copyright (2023) Dr. David A. Magezi
00002
00003 #ifndef DEF_ML_LOGGER
00004 #define DEF_ML_LOGGER
00005
00006 #include <string>
00007
00008 #include <dlib/logger.h>
00009
00010 namespace MachineLearning{
00011 class Logger{
00012 public:
00013     static void display_error_message(std::string error);
00014     static void display_info_message(std::string info);
00015     static void display_test_message();
00016     static void set_levels();
00017
00018 private:
00019     static dlib::logger error_log_;
00020     static dlib::logger info_log_;
00021     static dlib::logger test_log_;
00022 };
00023 }//namespace MachineLearning
00024
00025 #endif //DEF_ML_LOGGER
```

8.11 src/machine_learning/README.md File Reference

8.12 src/README.md File Reference

8.13 src/machine_learning/unet.cpp File Reference

```
#include "machine_learning/unet.h"
```


Namespaces

- namespace [MachineLearning](#)

8.14 src/machine_learning/unet.h File Reference

Classes

- class [MachineLearning::Unet](#)

Namespaces

- namespace [MachineLearning](#)

8.15 unet.h

[Go to the documentation of this file.](#)

```
00001 //Copyright 2023 David A. Magezi
00002 //Implementation of U-Net architecture by Ronneberger et al.
00003
00004 #ifndef DEF_ML_UNET
00005 #define DEF_ML_UNET
00006
00007 namespace MachineLearning{
00008     class Unet{
00009     public:
00010         Unet();
00011         ~Unet();
00012     };
00013 }//namespace MachineLearning
00014
00015 #endif //DEF_ML_UNET
```


Index

- ~Image
 - MachineLearning::Image, [15](#)
- ~Unet
 - MachineLearning::Unet, [16](#)
- ActiveGeometricShape, [11](#)
- ActiveGeometricShape::Ellipse, [13](#)
 - calculate_force_field, [13](#)
- ActiveGeometricShape::Ellipse::Image, [14](#)
 - height, [14](#)
 - width, [14](#)
 - x, [14](#)
- calculate_force_field
 - ActiveGeometricShape::Ellipse, [13](#)
- display_error_message
 - MachineLearning::Logger, [16](#)
- display_info_message
 - MachineLearning::Logger, [16](#)
- display_test_message
 - MachineLearning::Logger, [16](#)
- head_circumference.cpp
 - main, [18](#)
- height
 - ActiveGeometricShape::Ellipse::Image, [14](#)
- Image
 - MachineLearning::Image, [15](#)
- load_data
 - MachineLearning::Image, [15](#)
- machine learnig source code, [1](#)
- MachineLearning, [11](#)
- MachineLearning::Image, [14](#)
 - ~Image, [15](#)
 - Image, [15](#)
 - load_data, [15](#)
 - safe_append, [15](#)
 - set_test_status, [15](#)
- MachineLearning::Logger, [15](#)
 - display_error_message, [16](#)
 - display_info_message, [16](#)
 - display_test_message, [16](#)
 - set_levels, [16](#)
- MachineLearning::Unet, [16](#)
 - ~Unet, [16](#)
 - Unet, [16](#)
- main
 - head_circumference.cpp, [18](#)
- safe_append
 - MachineLearning::Image, [15](#)
- set_levels
 - MachineLearning::Logger, [16](#)
- set_test_status
 - MachineLearning::Image, [15](#)
- Source Files, [3](#)
 - src/active_geometric_shape/ellipse.cpp, [17](#)
 - src/active_geometric_shape/ellipse.h, [17](#), [18](#)
 - src/head_circumference.cpp, [18](#)
 - src/machine_learning/image.cpp, [18](#)
 - src/machine_learning/image.h, [19](#)
 - src/machine_learning/logger.cpp, [19](#)
 - src/machine_learning/logger.h, [20](#)
 - src/machine_learning/README.md, [20](#)
 - src/machine_learning/unet.cpp, [20](#)
 - src/machine_learning/unet.h, [21](#)
 - src/README.md, [20](#)
- Unet
 - MachineLearning::Unet, [16](#)
- width
 - ActiveGeometricShape::Ellipse::Image, [14](#)
- x
 - ActiveGeometricShape::Ellipse::Image, [14](#)