

Image Dataset Augmentor

v1.0

Generated by Doxygen 1.8.18

1 Augmentor Design Document	2
1.1 1. Introduction	2
1.2 2. Declarative APIs	2
1.2.1 2.1. Build with chain operations	2
1.2.2 2.2. Run the program	3
1.3 3. Template	3
1.3.1 3.1. Concept (in future)	3
1.4 4. Polymorphism	3
1.5 5. Compile-time programming	4
1.5.1 5.1. Different Implementations based on template parameters	4
1.5.1.1 5.1.1. Gaussian Blur Filter	4
1.5.1.2 5.1.2. Random Number Generator	5
1.6 6. Features	5
1.6.1 6.1. More realistic randomness	5
1.6.2 6.2. Fast Gaussian Blur	5
2 Namespace Index	6
2.1 Namespace List	6
3 Hierarchical Index	7
3.1 Class Hierarchy	7
4 Class Index	8
4.1 Class List	8
5 File Index	9
5.1 File List	9
6 Namespace Documentation	10
6.1 augmentorLib Namespace Reference	10
6.1.1 Function Documentation	11
6.1.1.1 convert2pixel()	11
6.1.2 Variable Documentation	11
6.1.2.1 LOWER_BOUND_PROB	11
6.1.2.2 NULL_SEED	11
6.1.2.3 UPPER_BOUND_PROB	11
6.2 jpegimageSTL Namespace Reference	12
6.3 jpegimageSTL::jpeg Namespace Reference	12
7 Class Documentation	13
7.1 augmentorLib::accumulator Struct Reference	13

7.1.1 Detailed Description	13
7.1.2 Member Typedef Documentation	14
7.1.2.1 _datatype	14
7.1.3 Constructor & Destructor Documentation	14
7.1.3.1 accumulator()	14
7.1.4 Member Function Documentation	14
7.1.4.1 add()	14
7.1.4.2 div()	14
7.1.4.3 shift()	15
7.1.5 Member Data Documentation	15
7.1.5.1 values	15
7.2 augmentorLib::Augmentor Class Reference	15
7.2.1 Detailed Description	16
7.2.2 Constructor & Destructor Documentation	16
7.2.2.1 Augmentor() [1/2]	16
7.2.2.2 ~Augmentor()	16
7.2.2.3 Augmentor() [2/2]	16
7.2.3 Member Function Documentation	17
7.2.3.1 blur() [1/2]	17
7.2.3.2 blur() [2/2]	17
7.2.3.3 crop()	18
7.2.3.4 flip()	18
7.2.3.5 invert()	20
7.2.3.6 pipeline()	20
7.2.3.7 random_erase() [1/2]	21
7.2.3.8 random_erase() [2/2]	21
7.2.3.9 rapid_blur()	22
7.2.3.10 resize() [1/4]	22
7.2.3.11 resize() [2/4]	23
7.2.3.12 resize() [3/4]	23
7.2.3.13 resize() [4/4]	24
7.2.3.14 rotate()	24
7.2.3.15 sample()	25
7.2.3.16 save()	25
7.2.3.17 zoom()	26
7.3 AugmentorTest Class Reference	26
7.3.1 Detailed Description	27
7.3.2 Constructor & Destructor Documentation	27
7.3.2.1 AugmentorTest()	27

7.3.2.2 ~AugmentorTest()	27
7.3.3 Member Function Documentation	27
7.3.3.1 SetUp()	27
7.3.3.2 TearDown()	28
7.3.4 Member Data Documentation	28
7.3.4.1 aug	28
7.3.4.2 img	28
7.4 augmentorLib::box_blur_filter_1D Struct Reference	28
7.4.1 Detailed Description	29
7.4.2 Constructor & Destructor Documentation	29
7.4.2.1 box_blur_filter_1D() [1/3]	29
7.4.2.2 box_blur_filter_1D() [2/3]	29
7.4.2.3 box_blur_filter_1D() [3/3]	29
7.4.3 Member Function Documentation	29
7.4.3.1 pseudo_gaussian_filter()	30
7.4.4 Member Data Documentation	30
7.4.4.1 length	30
7.5 augmentorLib::BoxBlurOperation< Image > Class Template Reference	30
7.5.1 Detailed Description	30
7.5.2 Constructor & Destructor Documentation	31
7.5.2.1 BoxBlurOperation() [1/2]	31
7.5.2.2 BoxBlurOperation() [2/2]	31
7.5.3 Member Function Documentation	31
7.5.3.1 perform()	31
7.6 augmentorLib::CropOperation< Image > Class Template Reference	32
7.6.1 Detailed Description	32
7.6.2 Constructor & Destructor Documentation	32
7.6.2.1 CropOperation() [1/2]	32
7.6.2.2 CropOperation() [2/2]	33
7.6.3 Member Function Documentation	33
7.6.3.1 perform()	33
7.7 augmentorLib::FastGaussianBlurOperation< Image > Class Template Reference	33
7.7.1 Detailed Description	34
7.7.2 Constructor & Destructor Documentation	34
7.7.2.1 FastGaussianBlurOperation()	34
7.7.3 Member Function Documentation	34
7.7.3.1 perform()	34
7.8 augmentorLib::FlipOperation< Image > Class Template Reference	35
7.8.1 Detailed Description	35

7.8.2 Constructor & Destructor Documentation	35
7.8.2.1 FlipOperation()	36
7.8.3 Member Function Documentation	36
7.8.3.1 perform()	36
7.9 augmentorLib::gaussian_blur_filter_1D< N, Static > Class Template Reference	36
7.9.1 Detailed Description	37
7.10 augmentorLib::gaussian_blur_filter_1D< N, false > Class Template Reference	37
7.10.1 Detailed Description	37
7.10.2 Constructor & Destructor Documentation	37
7.10.2.1 gaussian_blur_filter_1D() [1/2]	38
7.10.2.2 gaussian_blur_filter_1D() [2/2]	38
7.10.3 Member Function Documentation	38
7.10.3.1 operator[]()	38
7.10.3.2 size()	38
7.11 augmentorLib::gaussian_blur_filter_1D< N, true > Class Template Reference	38
7.11.1 Detailed Description	39
7.11.2 Constructor & Destructor Documentation	39
7.11.2.1 gaussian_blur_filter_1D() [1/2]	39
7.11.2.2 gaussian_blur_filter_1D() [2/2]	39
7.11.3 Member Function Documentation	39
7.11.3.1 operator[]()	40
7.11.3.2 size()	40
7.12 augmentorLib::GaussianBlurOperation< Image, Kernel > Class Template Reference	40
7.12.1 Detailed Description	40
7.12.2 Constructor & Destructor Documentation	41
7.12.2.1 GaussianBlurOperation() [1/2]	41
7.12.2.2 GaussianBlurOperation() [2/2]	41
7.12.3 Member Function Documentation	41
7.12.3.1 perform()	41
7.13 jpegimageSTL::jpeg::Image Class Reference	42
7.13.1 Detailed Description	42
7.13.2 Member Typedef Documentation	42
7.13.2.1 pixel_value_type	43
7.13.3 Constructor & Destructor Documentation	43
7.13.3.1 Image() [1/4]	43
7.13.3.2 Image() [2/4]	43
7.13.3.3 Image() [3/4]	44
7.13.3.4 ~Image()	44
7.13.3.5 Image() [4/4]	44

7.13.4 Member Function Documentation	44
7.13.4.1 getColorSpace()	45
7.13.4.2 getHeight()	45
7.13.4.3 getPixel()	45
7.13.4.4 getPixelSize()	45
7.13.4.5 getWidth()	46
7.13.4.6 resize()	46
7.13.4.7 save()	46
7.13.4.8 setPixel()	46
7.14 augmentorLib::image_size Struct Reference	47
7.14.1 Detailed Description	47
7.14.2 Member Data Documentation	47
7.14.2.1 height	47
7.14.2.2 width	48
7.15 augmentorLib::InvertOperation< Image > Class Template Reference	48
7.15.1 Detailed Description	48
7.15.2 Constructor & Destructor Documentation	48
7.15.2.1 InvertOperation()	48
7.15.3 Member Function Documentation	49
7.15.3.1 perform()	49
7.16 augmentorLib::Operation< Image > Class Template Reference	49
7.16.1 Detailed Description	50
7.16.2 Constructor & Destructor Documentation	50
7.16.2.1 Operation() [1/2]	50
7.16.2.2 ~Operation()	50
7.16.2.3 Operation() [2/2]	50
7.16.3 Member Function Documentation	51
7.16.3.1 operate_this_time()	51
7.16.3.2 perform() [1/2]	51
7.16.3.3 perform() [2/2]	51
7.16.3.4 uniform_random_number() [1/2]	52
7.16.3.5 uniform_random_number() [2/2]	52
7.17 augmentorLib::RandomEraseOperation< Image > Class Template Reference	52
7.17.1 Detailed Description	53
7.17.2 Constructor & Destructor Documentation	53
7.17.2.1 RandomEraseOperation()	53
7.17.3 Member Function Documentation	53
7.17.3.1 perform()	53
7.18 augmentorLib::ResizeOperation< Image > Class Template Reference	54

7.18.1 Detailed Description	54
7.18.2 Constructor & Destructor Documentation	54
7.18.2.1 ResizeOperation() [1/2]	55
7.18.2.2 ResizeOperation() [2/2]	55
7.18.3 Member Function Documentation	55
7.18.3.1 perform()	55
7.19 augmentorLib::rotate_range Struct Reference	56
7.19.1 Detailed Description	56
7.19.2 Member Data Documentation	56
7.19.2.1 max_rotate	56
7.19.2.2 min_rotate	56
7.20 augmentorLib::RotateOperation< Image > Class Template Reference	56
7.20.1 Detailed Description	57
7.20.2 Constructor & Destructor Documentation	57
7.20.2.1 RotateOperation() [1/2]	57
7.20.2.2 RotateOperation() [2/2]	57
7.20.3 Member Function Documentation	57
7.20.3.1 perform()	57
7.21 augmentorLib::StdoutOperation< Image > Class Template Reference	58
7.21.1 Detailed Description	58
7.21.2 Constructor & Destructor Documentation	58
7.21.2.1 StdoutOperation() [1/2]	59
7.21.2.2 StdoutOperation() [2/2]	59
7.21.3 Member Function Documentation	59
7.21.3.1 perform()	59
7.22 augmentorLib::UniformDistributionGenerator< DataType, IsReal > Class Template Reference	60
7.22.1 Detailed Description	60
7.23 augmentorLib::UniformDistributionGenerator< DataType, false > Class Template Reference	60
7.23.1 Detailed Description	60
7.23.2 Constructor & Destructor Documentation	61
7.23.2.1 UniformDistributionGenerator() [1/3]	61
7.23.2.2 ~UniformDistributionGenerator()	61
7.23.2.3 UniformDistributionGenerator() [2/3]	61
7.23.2.4 UniformDistributionGenerator() [3/3]	61
7.23.3 Member Function Documentation	62
7.23.3.1 operator>()	62
7.24 augmentorLib::UniformDistributionGenerator< DataType, true > Class Template Reference	62
7.24.1 Detailed Description	62
7.24.2 Constructor & Destructor Documentation	62

7.24.2.1 UniformDistributionGenerator() [1/3]	63
7.24.2.2 ~UniformDistributionGenerator()	63
7.24.2.3 UniformDistributionGenerator() [2/3]	63
7.24.2.4 UniformDistributionGenerator() [3/3]	63
7.24.3 Member Function Documentation	63
7.24.3.1 operator>()	64
7.25 augmentorLib::zoom_factor Struct Reference	64
7.25.1 Detailed Description	64
7.25.2 Member Data Documentation	64
7.25.2.1 max_factor	64
7.25.2.2 min_factor	64
7.26 augmentorLib::ZoomOperation< Image > Class Template Reference	65
7.26.1 Detailed Description	65
7.26.2 Constructor & Destructor Documentation	65
7.26.2.1 ZoomOperation() [1/2]	65
7.26.2.2 ZoomOperation() [2/2]	65
7.26.3 Member Function Documentation	66
7.26.3.1 perform()	66
8 File Documentation	68
8.1 /Users/gouthamreddykotapalle/Desktop/C++/docu/Augmentor Design Document.md File Reference	68
8.2 /Users/gouthamreddykotapalle/Desktop/C++/docu/Augmentor.cpp File Reference	68
8.2.1 Typedef Documentation	68
8.2.1.1 clocking	68
8.3 /Users/gouthamreddykotapalle/Desktop/C++/docu/Augmentor.h File Reference	69
8.4 /Users/gouthamreddykotapalle/Desktop/C++/docu/CMakeLists.txt File Reference	69
8.5 /Users/gouthamreddykotapalle/Desktop/C++/docu/filters.h File Reference	69
8.6 /Users/gouthamreddykotapalle/Desktop/C++/docu/jpeg.cpp File Reference	70
8.7 /Users/gouthamreddykotapalle/Desktop/C++/docu/jpeg.h File Reference	70
8.8 /Users/gouthamreddykotapalle/Desktop/C++/docu/main.cpp File Reference	70
8.8.1 Typedef Documentation	71
8.8.1.1 clocking	71
8.8.2 Function Documentation	71
8.8.2.1 main()	71
8.9 /Users/gouthamreddykotapalle/Desktop/C++/docu/Operation.cpp File Reference	71
8.10 /Users/gouthamreddykotapalle/Desktop/C++/docu/Operation.h File Reference	72
8.10.1 Macro Definition Documentation	73
8.10.1.1 HORIZONTAL	73
8.10.1.2 PI	73

8.10.1.3 VERTICAL	73
8.11 /Users/gouthamreddykotapalle/Desktop/C++/docu/unit_test.cpp File Reference	73
8.11.1 Function Documentation	74
8.11.1.1 main()	74
8.11.1.2 TEST_F() [1/5]	74
8.11.1.3 TEST_F() [2/5]	74
8.11.1.4 TEST_F() [3/5]	75
8.11.1.5 TEST_F() [4/5]	75
8.11.1.6 TEST_F() [5/5]	75
Index	76

Chapter 1

Augmentor Design Document

1.1 1. Introduction

`Augmentor` is a C++ library focused on image batch manipulation and processing. We would like to provide simple interface and excellent performance for users, so we have leveraged the following design ideas.

1.2 2. Declarative APIs

Similar as other data processing tools (e.g. Spark, Flink), our library also provides declarative APIs for users to build up their image manipulation pipelines. The advantage of declarative APIs is that the pipelines can be checked before the actual processing. The library evaluates the inputs of each operation, and make sure all of them are legal operations. For example, the input of `resize` operation cannot be negative. If invalid parameters are found, the program will be terminated in the building stage.

1.2.1 2.1. Build with chain operations

When users setup their pipelines, it is common to concatenate a set of operations to manipulate images. Thus, the pipeline builder of our library is designed to be chain-able. For example,

```
augmentor
.rotate(45,90,1)
.flip(HORIZONTAL, 1)
.crop(300, 300, true)
.resize(120,120,1)
.rapid_blur(5)
.invert(1);
```

In order to implement this style, we design the methods like
`Augmentor& Augmentor::some_operation(parameter param...);`

The `augmentor` has a vector to store the definitions of every operations, which looks like
`std::vector<std::unique_ptr< Operation<Image> » operations;`

The reason to use `unique_ptr` will be discussed more in the Section 4.

1.2.2 2.2. Run the program

After building up a pipeline, users can run it simply by calling `sample` method. The current implementation is straight forward: for each image, loop through every operation and perform the transform on the image. It looks something like

```
for (auto &operation : operations) {
    image = operation->perform(image);
}
```

Since the processing of images are independent of one another, parallel programming will be used in future to speed up the processing.

1.3 3. Template

There are many formats of images. They share the same APIs (e.g. `getPixel`, `setPixel`), but have different implementations. It is preferred to generalize the library to manipulate images in different formats, so Template is used in this library. Here is an example:

```
template<typename Image>
class Operation {
//
// private content...
//
public:
    virtual Image* perform(Image* image) = 0;
};
```

Currently, this library only supports JPEG images.

```
// Image actually means JpegImage
Operation<Image> operations;
```

In future, we will introduce more image formats, like png, bmp.

```
Operation<PngImage> PngOperations;
Operation<BmpImage> BmpOperations;
```

1.3.1 3.1. Concept (in future)

This library is built under the C++17 standard. In future, we would like to upgrade to C++20, where Concept is introduced. Since images, despite their formats, share the same interface (e.g. `getHeight`, `getWidth`, `getPixel`, `setPixel`), it is recommended to constrain an image class using Concept. A concept can make sure every image class share the same interface.

1.4 4. Polymorphism

This library uses the idea of Polymorphism to implement different operations. All operations inherit the base class `Operation`. It provides a few methods: (1) a `bool` randomizer to see if an operation take places this time; (2) a random number generator (between 0 and 1) to introduce randomness in each operation; and (3) a virtual function `perform` to ensure the same interface of its subclasses on image processing.

It looks like:

```
template<typename Image>
class Operation {
private:
    inline bool operate_this_time();
    inline _precision_type uniform_random_number();
```

```
public:
    virtual Image* perform(Image* image) = 0;
};
```

One subclass looks like

```
template<typename Image>
class ResizeOperation: public Operation<Image> {
public:
    Image* perform(Image* image) override;
};
```

A pipeline is made up of operations, and Augmentor stores the operations in a vector. We want to show the Augmentor's ownership of operations. There are two safe ways to show ownership: (1) a vector of objects, (2) a vector of unique pointers. Since a vector of base classes casts the objects of subclasses into base-class objects, we decide to use `unique_ptr` to show the ownership.

```
// Augmentor.h
class Augmentor {
    std::vector<std::unique_ptr< Operation<Image> » operations;
public:
    Augmentor& some_operation(parameters param...) {
        auto operation = std::make_unique<SomeOperation<Image>>(param);
        operations.push_back(std::move(operation));
        return *this;;
    }
}
```

The advantage for ownership lies in memory management. Before an Augmentor is going to be destroyed, the `unique_ptr` will first release the object it points to. Therefore, we can prevent any potential memory leaks in Augmentor class.

1.5 5. Compile-time programming

This library uses a lot of compile-time programming to optimize the code and performance. First of all, this library uses Template heavily. Since I have discussed Template in previous section, I will skip the basic usage of Template here. Instead, I will talk about how we can use Template to optimize code structure.

1.5.1 5.1. Different Implementations based on template parameters

1.5.1.1 5.1.1. Gaussian Blur Filter

Here I will show how to use template to make either static or dynamic filter when the GaussianBlue operation is built. The basic idea of blurring an image is to convolute between a filter and an image. The values of a filter can be either stored in array or vector. Storing in an array has advantages like fast accessing, but it requires users to specify the size before compile. Storing in a vector is more flexible, whose size is set in run-time, but it has relatively slow access. Therefore, I decide to use Template to integrate these two situations. The design looks like

```
template<unsigned N=0, bool Static = (N > 0) >
class gaussian_blur_filter_1D;
template<unsigned N>
class gaussian_blur_filter_1D<N, true> {
    double array[N];
public:
    explicit gaussian_blur_filter_1D(double sigma);
};
template<unsigned N>
class gaussian_blur_filter_1D<N, false> {
    std::vector<double> vector;
public:
    explicit gaussian_blur_filter_1D(double sigma, size_t n);
};
```

Either static and dynamic filters are created based on the constructor. Here is an example to build them.

```
// build a static filter with size of 5.
auto filter = gaussian_blur_filter_1D<5>(1.0);
// build a dynamic filter with size of 5.
auto filter = gaussian_blur_filter_1D(1.0, 5);
```

BlurOperation class wraps these two constructors into two member methods, so is the Augmentor class. Thus, users can build a blur operation either by `blur<5>(sigma)` or `blur(sigma, 5)`.

1.5.1.2 5.1.2. Random Number Generator

Here is another case to use template: allow a random number generator to output either integer or floating point numbers. In modern c++, `<random>` package is used to generate random numbers. It has two uniform number generators: `std::uniform_real_distribution` and `std::uniform_int_distribution`. Normally, if we want to build a generator like:

```
template <typename DataType>
class UniformDistributionGenerator<DataType> {
public:
    inline DataType operator()();
}
```

We may include both uniform number generators mentioned above as members, and call them based on datatype. However, we can slightly modify the template and split this template into two implementations:

```
template <typename DataType, bool IsReal = std::is_floating_point<DataType>::value>
class UniformDistributionGenerator;
class UniformDistributionGenerator<DataType, true> {
    std::uniform_real_distribution<DataType> distribution;
}
class UniformDistributionGenerator<DataType, false> {
    std::uniform_int_distribution<DataType> distribution;
}
```

In this way, there is no overhead from redundant members. Also, the performance is better since we don't have to do if-else evaluation when calling `operator()`. The performance difference here may not be significant due to the simple function here, but this idea can be applied to more complex design.

1.6 6. Features

There are other design features that distinguish our library from others.

1.6.1 6.1. More realistic randomness

Many image processing libraries (e.g. PIL) use `rand()` to generate random numbers, but this method may cause a few issues. please see this [Q&A](#). On the contrary, our library uses `<random>` to achieve more realistic randomness. We use current timestamp as seed to create a generator, and then use `uniform_distribution` to output random numbers. Our library should result in better randomness than others.

1.6.2 6.2. Fast Gaussian Blur

This library implements some optimized algorithm to increase performance. One example is the [Fast Gaussian Blur](#). Since Gaussian Blur is expensive, whose complexity should be at least $O(N * r)$, where N is the area of an image and r is the size of a filter. However, research have found that multiple Box Blurs can approximate the result of Gaussian Blur, and the complexity of a Box Blur can be as low as $O(N)$. Therefore, this library decides to implement the fast Gaussian Blur to increase the performance. The details can be found in the `Operation.h` file.

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

augmentorLib	10
jpegimageSTL	12
jpegimageSTL::jpeg	12

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

augmentorLib::accumulator	13
augmentorLib::Augmentor	15
augmentorLib::box_blur_filter_1D	28
augmentorLib::gaussian_blur_filter_1D< N, Static >	36
augmentorLib::gaussian_blur_filter_1D< 0 >	36
augmentorLib::gaussian_blur_filter_1D< N, false >	37
augmentorLib::gaussian_blur_filter_1D< N, true >	38
jpegimageSTL::jpeg::Image	42
augmentorLib::image_size	47
augmentorLib::Operation< Image >	49
augmentorLib::BoxBlurOperation< Image >	30
augmentorLib::CropOperation< Image >	32
augmentorLib::FastGaussianBlurOperation< Image >	33
augmentorLib::FlipOperation< Image >	35
augmentorLib::GaussianBlurOperation< Image, Kernel >	40
augmentorLib::InvertOperation< Image >	48
augmentorLib::RandomEraseOperation< Image >	52
augmentorLib::ResizeOperation< Image >	54
augmentorLib::RotateOperation< Image >	56
augmentorLib::StdoutOperation< Image >	58
augmentorLib::ZoomOperation< Image >	65
augmentorLib::rotate_range	56
Test	
AugmentorTest	26
augmentorLib::UniformDistributionGenerator< DataType, IsReal >	60
augmentorLib::UniformDistributionGenerator< _precision_type >	60
augmentorLib::UniformDistributionGenerator< DataType, false >	60
augmentorLib::UniformDistributionGenerator< DataType, true >	62
augmentorLib::UniformDistributionGenerator< pixel_value_type >	60
augmentorLib::UniformDistributionGenerator< size_t >	60
augmentorLib::zoom_factor	64

Chapter 4

Class Index

4.1 Class List

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

augmentorLib::accumulator	13
augmentorLib::Augmentor	15
AugmentorTest	26
augmentorLib::box_blur_filter_1D	28
augmentorLib::BoxBlurOperation< Image >	30
augmentorLib::CropOperation< Image >	32
augmentorLib::FastGaussianBlurOperation< Image >	33
augmentorLib::FlipOperation< Image >	35
augmentorLib::gaussian_blur_filter_1D< N, Static >	36
augmentorLib::gaussian_blur_filter_1D< N, false >	37
augmentorLib::gaussian_blur_filter_1D< N, true >	38
augmentorLib::GaussianBlurOperation< Image, Kernel >	40
jpegimageSTL::jpeg::Image	42
augmentorLib::image_size	47
augmentorLib::InvertOperation< Image >	48
augmentorLib::Operation< Image >	49
augmentorLib::RandomEraseOperation< Image >	52
augmentorLib::ResizeOperation< Image >	54
augmentorLib::rotate_range	56
augmentorLib::RotateOperation< Image >	56
augmentorLib::StdoutOperation< Image >	58
augmentorLib::UniformDistributionGenerator< DataType, IsReal >	60
augmentorLib::UniformDistributionGenerator< DataType, false >	60
augmentorLib::UniformDistributionGenerator< DataType, true >	62
augmentorLib::zoom_factor	64
augmentorLib::ZoomOperation< Image >	65

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

/Users/gouthamreddykotapalle/Desktop/C++/docu/ Augmentor.cpp	68
/Users/gouthamreddykotapalle/Desktop/C++/docu/ Augmentor.h	69
/Users/gouthamreddykotapalle/Desktop/C++/docu/ filters.h	69
/Users/gouthamreddykotapalle/Desktop/C++/docu/ jpeg.cpp	70
/Users/gouthamreddykotapalle/Desktop/C++/docu/ jpeg.h	70
/Users/gouthamreddykotapalle/Desktop/C++/docu/ main.cpp	70
/Users/gouthamreddykotapalle/Desktop/C++/docu/ Operation.cpp	71
/Users/gouthamreddykotapalle/Desktop/C++/docu/ Operation.h	72
/Users/gouthamreddykotapalle/Desktop/C++/docu/ unit_test.cpp	73

Chapter 6

Namespace Documentation

6.1 augmentorLib Namespace Reference

Classes

- struct [accumulator](#)
- class [Augmentor](#)
- struct [box_blur_filter_1D](#)
- class [BoxBlurOperation](#)
- class [CropOperation](#)
- class [FastGaussianBlurOperation](#)
- class [FlipOperation](#)
- class [gaussian_blur_filter_1D](#)
- class [gaussian_blur_filter_1D< N, false >](#)
- class [gaussian_blur_filter_1D< N, true >](#)
- class [GaussianBlurOperation](#)
- struct [image_size](#)
- class [InvertOperation](#)
- class [Operation](#)
- class [RandomEraseOperation](#)
- class [ResizeOperation](#)
- struct [rotate_range](#)
- class [RotateOperation](#)
- class [StdoutOperation](#)
- class [UniformDistributionGenerator](#)
- class [UniformDistributionGenerator< DataType, false >](#)
- class [UniformDistributionGenerator< DataType, true >](#)
- struct [zoom_factor](#)
- class [ZoomOperation](#)

Functions

- void [convert2pixel](#) (const std::vector< double > &src, std::vector< uint8_t > &target, size_t n)

Variables

- const double `LOWER_BOUND_PROB` = 0.0
- const double `UPPER_BOUND_PROB` = 1.0
- const unsigned `NULL_SEED` = 0

6.1.1 Function Documentation

6.1.1.1 `convert2pixel()`

```
void augmentorLib::convert2pixel (
    const std::vector< double > & src,
    std::vector< uint8_t > & target,
    size_t n ) [inline]
```

Definition at line 564 of file Operation.h.

6.1.2 Variable Documentation

6.1.2.1 `LOWER_BOUND_PROB`

```
const double augmentorLib::LOWER_BOUND_PROB = 0.0
```

Definition at line 24 of file Operation.h.

6.1.2.2 `NULL_SEED`

```
const unsigned augmentorLib::NULL_SEED = 0
```

Definition at line 26 of file Operation.h.

6.1.2.3 `UPPER_BOUND_PROB`

```
const double augmentorLib::UPPER_BOUND_PROB = 1.0
```

Definition at line 25 of file Operation.h.

6.2 jpegimageSTL Namespace Reference

Namespaces

- [jpeg](#)

6.3 jpegimageSTL::jpeg Namespace Reference

Classes

- class [Image](#)

Chapter 7

Class Documentation

7.1 augmentorLib::accumulator Struct Reference

```
#include <Operation.h>
```

Public Types

- typedef uint64_t [_datatype](#)

Public Member Functions

- [accumulator](#) (size_t n)
- template<typename Value >
void [add](#) (std::vector< Value > &&val)
- template<typename Value >
void [shift](#) (std::vector< Value > &&del, std::vector< Value > &&[add](#))
- template<typename ReturnType = u_int8_t>
std::vector< ReturnType > [div](#) ([_datatype](#) denominator)

Public Attributes

- std::vector< [_datatype](#) > [values](#)

7.1.1 Detailed Description

Definition at line 621 of file Operation.h.

7.1.2 Member Typedef Documentation

7.1.2.1 `_datatype`

```
typedef uint64_t augmentorLib::accumulator::_datatype
```

Definition at line 622 of file Operation.h.

7.1.3 Constructor & Destructor Documentation

7.1.3.1 `accumulator()`

```
augmentorLib::accumulator::accumulator (  
    size_t n ) [inline], [explicit]
```

Definition at line 625 of file Operation.h.

7.1.4 Member Function Documentation

7.1.4.1 `add()`

```
template<typename Value >  
void augmentorLib::accumulator::add (  
    std::vector< Value > && val ) [inline]
```

Definition at line 628 of file Operation.h.

7.1.4.2 `div()`

```
template<typename ReturnType = u_int8_t>  
std::vector<ReturnType> augmentorLib::accumulator::div (  
    _datatype denominator ) [inline]
```

Definition at line 645 of file Operation.h.

7.1.4.3 shift()

```
template<typename Value >
void augmentorLib::accumulator::shift (
    std::vector< Value > && del,
    std::vector< Value > && add ) [inline]
```

Definition at line 636 of file Operation.h.

7.1.5 Member Data Documentation

7.1.5.1 values

```
std::vector<_datatype> augmentorLib::accumulator::values
```

Definition at line 623 of file Operation.h.

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

- /Users/gouthamreddykotapalle/Desktop/C++/docu/[Operation.h](#)

7.2 augmentorLib::Augmentor Class Reference

```
#include <Augmentor.h>
```

Public Member Functions

- [Augmentor](#) ()=default
Default Constructor.
- [~Augmentor](#) ()=default
- [Augmentor](#) (const std::string &in_path, const std::string &out_path)
- [Augmentor](#) & [resize](#) ([image_size](#) lower, [image_size](#) upper, double prob=1)
- [Augmentor](#) & [resize](#) ([image_size](#) new_size, double prob=1)
- [Augmentor](#) & [resize](#) (size_t lower_height, size_t lower_width, size_t upper_height, size_t upper_width, double prob=1)
- [Augmentor](#) & [resize](#) (size_t height, size_t width, double prob=1)
- [Augmentor](#) & [crop](#) (int height, int width, bool center, double prob=1)
- [Augmentor](#) & [zoom](#) (double min_factor=1.0, double max_factor=1.0, double prob=1)
- [Augmentor](#) & [rotate](#) (int min_degree, int max_degree, double prob=1)
- [Augmentor](#) & [invert](#) (double prob=1)
- template<int K = 5>
[Augmentor](#) & [blur](#) (double sigma, double prob=1)
- [Augmentor](#) & [blur](#) (double sigma, size_t kernel_size, double prob=1)
- [Augmentor](#) & [rapid_blur](#) (const double sigma, const unsigned int passes=3, double prob=1)
- [Augmentor](#) & [pipeline](#) ()
- [Augmentor](#) & [random_erase](#) ([image_size](#) lower_mask_size, [image_size](#) upper_mask_size, double prob=1)
- [Augmentor](#) & [random_erase](#) ([image_size](#) mask_size, double prob=1)
- [Augmentor](#) & [flip](#) (const std::string &type, double prob=1)
- void [sample](#) (size_t size)

Static Public Member Functions

- static void `save` (const std::string &fileName, `Image` *image, int quality=95)

7.2.1 Detailed Description

This is the `Augmentor` Class.

This is the main class of the library, an instance of which the user would create for sampling images

Definition at line 18 of file `Augmentor.h`.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 `Augmentor()` [1/2]

```
augmentorLib::Augmentor::Augmentor ( ) [default]
```

Default Constructor.

7.2.2.2 `~Augmentor()`

```
augmentorLib::Augmentor::~~Augmentor ( ) [default]
```

7.2.2.3 `Augmentor()` [2/2]

```
augmentorLib::Augmentor::Augmentor (
    const std::string & in_path,
    const std::string & out_path ) [explicit]
```

Load an image file. Currently only accepts RGB colorspace images.

Parameters

<i>in_path</i>	- Input directory path
<i>out_path</i>	- Output directory path

Definition at line 8 of file Augmentor.cpp.

7.2.3 Member Function Documentation

7.2.3.1 blur() [1/2]

```
template<int K = 5>
Augmentor& augmentorLib::Augmentor::blur (
    double sigma,
    double prob = 1 ) [inline]
```

Blur

Blurs the image based on the sigma value

Template Parameters

<i>K</i>	Kernel value for Gaussian distribution
----------	--

Parameters

<i>sigma</i>	Sigma value for Gaussian distribution
<i>prob</i>	probability of performing the resize operation

Returns

A reference to the [Augmentor](#) object

Definition at line 124 of file Augmentor.h.

7.2.3.2 blur() [2/2]

```
Augmentor & augmentorLib::Augmentor::blur (
    double sigma,
    size_t kernel_size,
    double prob = 1 )
```

Blur

Blurs the image based on the sigma value

Parameters

<i>sigma</i>	Sigma value for Gaussian distribution
<i>kernel_size</i>	Kernel size for Gaussian distribution
<i>prob</i>	probability of performing the resize operation

Returns

A reference to the [Augmentor](#) object

Definition at line 131 of file Augmentor.cpp.

7.2.3.3 crop()

```
Augmentor & augmentorLib::Augmentor::crop (  
    int height,  
    int width,  
    bool center,  
    double prob = 1 )
```

Crop the image

Crop based on current width and height

Parameters

<i>height</i>	- new height of the augmented image
<i>width</i>	- new width of the augmented image
<i>center</i>	- boolean value to crop by using the center as pivot or not
<i>prob</i>	- probability of performing the resize operation

Returns

A reference to the [Augmentor](#) object

Definition at line 64 of file Augmentor.cpp.

7.2.3.4 flip()

```
Augmentor & augmentorLib::Augmentor::flip (  
    const std::string & type,  
    double prob = 1 )
```

Flip

Flips the image either vertically or horizontally

Parameters

<i>type</i>	Takes any of the two options - VERTICAL or HORIZONTAL
<i>prob</i>	probability of performing the resize operation

Returns

A reference to the [Augmentor](#) object

Definition at line 156 of file Augmentor.cpp.

7.2.3.5 invert()

```
Augmentor & augmentorLib::Augmentor::invert (
    double prob = 1 )
```

Invert the image Inverts the colors in the image

Parameters

<i>prob</i>	probability of performing the resize operation
-------------	--

Returns

A reference to the [Augmentor](#) object

Definition at line 88 of file Augmentor.cpp.

7.2.3.6 pipeline()

```
Augmentor & augmentorLib::Augmentor::pipeline ( )
```

Pipeline Creates an input image array to operate on

Returns

A reference to the [Augmentor](#) object

Definition at line 19 of file Augmentor.cpp.

7.2.3.7 random_erase() [1/2]

```
Augmentor & augmentorLib::Augmentor::random_erase (
    image_size lower_mask_size,
    image_size upper_mask_size,
    double prob = 1 )
```

Random erase

Randomly erase a part of the image based on a mask size selected in random from the range specified

Parameters

<i>lower_mask_size</i>	lower mask size used as { height, width }
<i>upper_mask_size</i>	upper mask size used as { height, width }
<i>prob</i>	probability of performing the resize operation

Attention

pixels are lost in the output image

Returns

A reference to the [Augmentor](#) object

Definition at line 144 of file Augmentor.cpp.

7.2.3.8 random_erase() [2/2]

```
Augmentor & augmentorLib::Augmentor::random_erase (
    image_size mask_size,
    double prob = 1 )
```

Random erase

Randomly erase a part of the image based on a specified mask size

Parameters

<i>mask_size</i>	mask size used as { height, width }
<i>prob</i>	probability of performing the resize operation

Attention

pixels are lost in the output image

Returns

A reference to the [Augmentor](#) object

Definition at line 150 of file Augmentor.cpp.

7.2.3.9 rapid_blur()

```
Augmentor & augmentorLib::Augmentor::rapid_blur (
    const double sigma,
    const unsigned int passes = 3,
    double prob = 1 )
```

Definition at line 137 of file Augmentor.cpp.

7.2.3.10 resize() [1/4]

```
Augmentor & augmentorLib::Augmentor::resize (
    image\_size lower,
    image\_size upper,
    double prob = 1 )
```

Resize the image

expand or shrink based on a size selected in random from the range specified

Parameters

<i>lower</i>	- lower bound on the dimensions of resize
<i>upper</i>	- upper bound on the dimensions of resize
<i>prob</i>	- probability of performing the resize operation

Returns

A reference to the [Augmentor](#) object

See also

[resize\(image_size new_size, double prob=1\)](#) [resize\(size_t lower_height, size_t lower_width, size_t upper_height, size_t upper_width, double prob=1\)](#) [resize\(size_t height, size_t width, double prob=1\)](#)

Definition at line 36 of file Augmentor.cpp.

7.2.3.11 `resize()` [2/4]

```
Augmentor & augmentorLib::Augmentor::resize (  
    image_size new_size,  
    double prob = 1 )
```

Resize the image

expand or shrink based on current width

Parameters

<i>new_size</i>	- new size of the augmented image
<i>prob</i>	- probability of performing the resize operation

Returns

A reference to the [Augmentor](#) object

Definition at line 42 of file Augmentor.cpp.

7.2.3.12 `resize()` [3/4]

```
Augmentor & augmentorLib::Augmentor::resize (  
    size_t height,  
    size_t width,  
    double prob = 1 )
```

Resize the image

expand or shrink based on current width

Parameters

<i>new_size</i>	- new size of the augmented image
<i>prob</i>	- probability of performing the resize operation

Returns

A reference to the [Augmentor](#) object

Definition at line 56 of file Augmentor.cpp.

7.2.3.13 resize() [4/4]

```
Augmentor & augmentorLib::Augmentor::resize (
    size_t lower_height,
    size_t lower_width,
    size_t upper_height,
    size_t upper_width,
    double prob = 1 )
```

Resize the image

expand or shrink based on a size selected in random from the range specified

Parameters

<i>lower_height</i>	- min height for the range
<i>lower_width</i>	- min width of the range
<i>upper_height</i>	- max height of the range
<i>upper_width</i>	- max width of the range
<i>prob</i>	- probability of performing the resize operation

Returns

A reference to the [Augmentor](#) object

Definition at line 48 of file Augmentor.cpp.

7.2.3.14 rotate()

```
Augmentor & augmentorLib::Augmentor::rotate (
    int min_degree,
    int max_degree,
    double prob = 1 )
```

Rotate the image

Rotate based on current width and height based on a angle selected in random from the range specified

Parameters

<i>min_factor</i>	- minimum zoom factor of range
<i>max_factor</i>	- maximum zoom factor of range
<i>prob</i>	- probability of performing the resize operation

Returns

A reference to the [Augmentor](#) object

Definition at line 80 of file Augmentor.cpp.

7.2.3.15 sample()

```
void augmentorLib::Augmentor::sample (  
    size_t size )
```

Sample

creates the specified number of augmented images

Parameters

<i>size</i>	number of augmented images to specify
-------------	---------------------------------------

Definition at line 94 of file Augmentor.cpp.

7.2.3.16 save()

```
void augmentorLib::Augmentor::save (  
    const std::string & fileName,  
    Image * image,  
    int quality = 95 ) [static]
```

Save current version of image into a file specified, with the default/specified quality (0-100)

Parameters

<i>fileName</i>	- Name of the augmented output file
<i>image</i>	- an image of type Image
<i>quality</i>	- quality of the output image

Returns

A reference to the [Augmentor](#) object

Definition at line 15 of file Augmentor.cpp.

7.2.3.17 zoom()

```
Augmentor & augmentorLib::Augmentor::zoom (
    double min_factor = 1.0,
    double max_factor = 1.0,
    double prob = 1 )
```

Zoom the image

Zoom based on current width and height based on a factor selected in random from the range specified

Parameters

<i>min_factor</i>	- minimum zoom factor of range
<i>max_factor</i>	- maximum zoom factor of range
<i>prob</i>	- probability of performing the resize operation

Returns

A reference to the [Augmentor](#) object

Definition at line 72 of file Augmentor.cpp.

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

- /Users/gouthamreddykotapalle/Desktop/C++/docu/[Augmentor.h](#)
- /Users/gouthamreddykotapalle/Desktop/C++/docu/[Augmentor.cpp](#)

7.3 AugmentorTest Class Reference

Inherits Test.

Protected Member Functions

- [AugmentorTest](#) ()
- virtual [~AugmentorTest](#) ()
- virtual void [SetUp](#) ()
- virtual void [TearDown](#) ()

Protected Attributes

- [augmentorLib::Augmentor](#) * aug
- [Image](#) * img

7.3.1 Detailed Description

Definition at line 9 of file unit_test.cpp.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 AugmentorTest()

```
AugmentorTest::AugmentorTest ( ) [inline], [protected]
```

Definition at line 15 of file unit_test.cpp.

7.3.2.2 ~AugmentorTest()

```
virtual AugmentorTest::~~AugmentorTest ( ) [inline], [protected], [virtual]
```

Definition at line 19 of file unit_test.cpp.

7.3.3 Member Function Documentation

7.3.3.1 SetUp()

```
virtual void AugmentorTest::SetUp ( ) [inline], [protected], [virtual]
```

Definition at line 22 of file unit_test.cpp.

7.3.3.2 TearDown()

```
virtual void AugmentorTest::TearDown ( ) [inline], [protected], [virtual]
```

Definition at line 27 of file unit_test.cpp.

7.3.4 Member Data Documentation

7.3.4.1 aug

```
augmentorLib::Augmentor* AugmentorTest::aug [protected]
```

Definition at line 12 of file unit_test.cpp.

7.3.4.2 img

```
Image* AugmentorTest::img [protected]
```

Definition at line 13 of file unit_test.cpp.

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

- /Users/gouthamreddykotapalle/Desktop/C++/docu/[unit_test.cpp](#)

7.4 augmentorLib::box_blur_filter_1D Struct Reference

```
#include <filters.h>
```

Public Member Functions

- [box_blur_filter_1D](#) ()=delete
- [box_blur_filter_1D](#) (size_t length)
- [box_blur_filter_1D](#) (const [box_blur_filter_1D](#) &other)

Static Public Member Functions

- static std::vector< [box_blur_filter_1D](#) > [pseudo_gaussian_filter](#) (double sigma, unsigned passes)

Public Attributes

- const size_t [length](#)

7.4.1 Detailed Description

Definition at line 95 of file filters.h.

7.4.2 Constructor & Destructor Documentation

7.4.2.1 box_blur_filter_1D() [1/3]

```
augmentorLib::box_blur_filter_1D::box_blur_filter_1D ( ) [delete]
```

7.4.2.2 box_blur_filter_1D() [2/3]

```
augmentorLib::box_blur_filter_1D::box_blur_filter_1D (  
    size_t length ) [inline], [explicit]
```

Definition at line 100 of file filters.h.

7.4.2.3 box_blur_filter_1D() [3/3]

```
augmentorLib::box_blur_filter_1D::box_blur_filter_1D (  
    const box\_blur\_filter\_1D & other ) [inline]
```

Definition at line 104 of file filters.h.

7.4.3 Member Function Documentation

7.4.3.1 pseudo_gaussian_filter()

```
static std::vector<box_blur_filter_1D> augmentorLib::box_blur_filter_1D::pseudo_gaussian_filter (
    double sigma,
    unsigned passes ) [inline], [static]
```

Definition at line 107 of file filters.h.

7.4.4 Member Data Documentation

7.4.4.1 length

```
const size_t augmentorLib::box_blur_filter_1D::length
```

Definition at line 96 of file filters.h.

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

- [/Users/gouthamreddykotapalle/Desktop/C++/docu/filters.h](#)

7.5 augmentorLib::BoxBlurOperation< Image > Class Template Reference

```
#include <Operation.h>
```

Inherits [augmentorLib::Operation< Image >](#).

Public Member Functions

- [BoxBlurOperation](#) (const size_t n, double prob=UPPER_BOUND_PROB, unsigned seed=NULL_SEED)
- [BoxBlurOperation](#) (const [box_blur_filter_1D](#) filter, double prob=UPPER_BOUND_PROB, unsigned seed=NULL_SEED)
- [Image * perform](#) ([Image *image](#)) override

Additional Inherited Members

7.5.1 Detailed Description

```
template<typename Image>
class augmentorLib::BoxBlurOperation< Image >
```

Definition at line 287 of file Operation.h.

7.5.2 Constructor & Destructor Documentation

7.5.2.1 BoxBlurOperation() [1/2]

```
template<typename Image >
augmentorLib::BoxBlurOperation< Image >::BoxBlurOperation (
    const size_t n,
    double prob = UPPER_BOUND_PROB,
    unsigned seed = NULL_SEED ) [inline], [explicit]
```

Definition at line 291 of file Operation.h.

7.5.2.2 BoxBlurOperation() [2/2]

```
template<typename Image >
augmentorLib::BoxBlurOperation< Image >::BoxBlurOperation (
    const box_blur_filter_1D filter,
    double prob = UPPER_BOUND_PROB,
    unsigned seed = NULL_SEED ) [inline], [explicit]
```

Definition at line 295 of file Operation.h.

7.5.3 Member Function Documentation

7.5.3.1 perform()

```
template<typename Image >
Image * augmentorLib::BoxBlurOperation< Image >::perform (
    Image * image ) [override], [virtual]
```

Perform function that is called to invoke a particular operation

Parameters

<i>image</i>	Image to perform an operation on
--------------	----------------------------------

Returns

A pointer to an image object

Implements [augmentorLib::Operation< Image >](#).

Definition at line 655 of file Operation.h.

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

- /Users/gouthamreddykotapalle/Desktop/C++/docu/[Operation.h](#)

7.6 augmentorLib::CropOperation< Image > Class Template Reference

```
#include <Operation.h>
```

Inherits [augmentorLib::Operation< Image >](#).

Public Member Functions

- [CropOperation](#) ()=delete
- [CropOperation](#) (image_size size, bool center, double prob=UPPER_BOUND_PROB, unsigned seed=NULL_SEED)
- Image * [perform](#) (Image *image) override

Additional Inherited Members

7.6.1 Detailed Description

```
template<typename Image>  
class augmentorLib::CropOperation< Image >
```

Definition at line 202 of file Operation.h.

7.6.2 Constructor & Destructor Documentation

7.6.2.1 CropOperation() [1/2]

```
template<typename Image >  
augmentorLib::CropOperation< Image >::CropOperation ( ) [delete]
```


7.6.2.2 CropOperation() [2/2]

```
template<typename Image >
augmentorLib::CropOperation< Image >::CropOperation (
    image_size size,
    bool center,
    double prob = UPPER_BOUND_PROB,
    unsigned seed = NULL_SEED ) [inline], [explicit]
```

Definition at line 210 of file Operation.h.

7.6.3 Member Function Documentation

7.6.3.1 perform()

```
template<typename Image >
Image * augmentorLib::CropOperation< Image >::perform (
    Image * image ) [override], [virtual]
```

Perform function that is called to invoke a particular operation

Parameters

<i>image</i>	Image to perform an operation on
--------------	----------------------------------

Returns

A pointer to an image object

Implements [augmentorLib::Operation< Image >](#).

Definition at line 427 of file Operation.h.

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

- /Users/gouthamreddykotapalle/Desktop/C++/docu/[Operation.h](#)

7.7 augmentorLib::FastGaussianBlurOperation< Image > Class Template Reference

```
#include <Operation.h>
```

Inherits [augmentorLib::Operation< Image >](#).

Public Member Functions

- [FastGaussianBlurOperation](#) (const double sigma, const unsigned int passes, double prob=[UPPER_BOUND_PROB](#), unsigned seed=[NULL_SEED](#))
- Image * [perform](#) (Image *image) override

Additional Inherited Members

7.7.1 Detailed Description

```
template<typename Image>
class augmentorLib::FastGaussianBlurOperation< Image >
```

Definition at line 302 of file Operation.h.

7.7.2 Constructor & Destructor Documentation

7.7.2.1 FastGaussianBlurOperation()

```
template<typename Image >
augmentorLib::FastGaussianBlurOperation< Image >::FastGaussianBlurOperation (
    const double sigma,
    const unsigned int passes,
    double prob = UPPER\_BOUND\_PROB,
    unsigned seed = NULL\_SEED ) [inline], [explicit]
```

Definition at line 307 of file Operation.h.

7.7.3 Member Function Documentation

7.7.3.1 perform()

```
template<typename Image >
Image * augmentorLib::FastGaussianBlurOperation< Image >::perform (
    Image * image ) [override], [virtual]
```

Perform function that is called to invoke a particular operation

Parameters

<i>image</i>	Image to perform an operation on
--------------	----------------------------------

Returns

A pointer to an image object

Implements [augmentorLib::Operation< Image >](#).

Definition at line 708 of file Operation.h.

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

- /Users/gouthamreddykotapalle/Desktop/C++/docu/[Operation.h](#)

7.8 augmentorLib::FlipOperation< Image > Class Template Reference

```
#include <Operation.h>
```

Inherits [augmentorLib::Operation< Image >](#).

Public Member Functions

- [FlipOperation](#) (const std::string &type, double prob=UPPER_BOUND_PROB, unsigned seed=NULL_SEED)
- Image * [perform](#) (Image *image) override

Additional Inherited Members

7.8.1 Detailed Description

```
template<typename Image>  
class augmentorLib::FlipOperation< Image >
```

Definition at line 343 of file Operation.h.

7.8.2 Constructor & Destructor Documentation

7.8.2.1 FlipOperation()

```
template<typename Image >
augmentorLib::FlipOperation< Image >::FlipOperation (
    const std::string & type,
    double prob = UPPER_BOUND_PROB,
    unsigned seed = NULL_SEED ) [inline], [explicit]
```

Definition at line 347 of file Operation.h.

7.8.3 Member Function Documentation

7.8.3.1 perform()

```
template<typename Image >
Image * augmentorLib::FlipOperation< Image >::perform (
    Image * image ) [override], [virtual]
```

Perform function that is called to invoke a particular operation

Parameters

<i>image</i>	Image to perform an operaion on
--------------	---------------------------------

Returns

A pointer to an image object

Implements [augmentorLib::Operation< Image >](#).

Definition at line 356 of file Operation.h.

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

- [/Users/gouthamreddykotapalle/Desktop/C++/docu/Operation.h](#)

7.9 augmentorLib::gaussian_blur_filter_1D< N, Static > Class Template Reference

```
#include <filters.h>
```

7.9.1 Detailed Description

```
template<unsigned N = 0, bool Static = (N > 0)>
class augmentorLib::gaussian_blur_filter_1D< N, Static >
```

Definition at line 15 of file filters.h.

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

- [/Users/gouthamreddykotapalle/Desktop/C++/docu/filters.h](#)

7.10 augmentorLib::gaussian_blur_filter_1D< N, false > Class Template Reference

```
#include <filters.h>
```

Public Member Functions

- [gaussian_blur_filter_1D](#) ()=delete
- [gaussian_blur_filter_1D](#) (double sigma, size_t n)
- double [operator\[\]](#) (size_t i)
- size_t [size](#) ()

7.10.1 Detailed Description

```
template<unsigned N>
class augmentorLib::gaussian_blur_filter_1D< N, false >
```

Gaussian blur filter class

False Version

Template Parameters

<i>N</i>	Denotes the Kernel value for the blur operation
----------	---

Definition at line 61 of file filters.h.

7.10.2 Constructor & Destructor Documentation

7.10.2.1 `gaussian_blur_filter_1D()` [1/2]

```
template<unsigned N>
augmentorLib::gaussian_blur_filter_1D< N, false >::gaussian_blur_filter_1D ( ) [delete]
```

7.10.2.2 `gaussian_blur_filter_1D()` [2/2]

```
template<unsigned N>
augmentorLib::gaussian_blur_filter_1D< N, false >::gaussian_blur_filter_1D (
    double sigma,
    size_t n ) [inline], [explicit]
```

Definition at line 68 of file filters.h.

7.10.3 Member Function Documentation

7.10.3.1 `operator[]()`

```
template<unsigned N>
double augmentorLib::gaussian_blur_filter_1D< N, false >::operator[] (
    size_t i ) [inline]
```

Definition at line 86 of file filters.h.

7.10.3.2 `size()`

```
template<unsigned N>
size_t augmentorLib::gaussian_blur_filter_1D< N, false >::size ( ) [inline]
```

Definition at line 90 of file filters.h.

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

- `/Users/gouthamreddykotapalle/Desktop/C++/docu/filters.h`

7.11 `augmentorLib::gaussian_blur_filter_1D< N, true >` Class Template Reference

```
#include <filters.h>
```

Public Member Functions

- [gaussian_blur_filter_1D](#) ()=delete
- [gaussian_blur_filter_1D](#) (const double sigma)
- double [operator\[\]](#) (size_t i)
- size_t [size](#) ()

7.11.1 Detailed Description

```
template<unsigned N>
class augmentorLib::gaussian_blur_filter_1D< N, true >
```

Gaussian blur filter class

True Version

Template Parameters

<i>N</i>	Denotes the Kernel value for the blur operation
----------	---

Definition at line 22 of file filters.h.

7.11.2 Constructor & Destructor Documentation

7.11.2.1 [gaussian_blur_filter_1D\(\)](#) [1/2]

```
template<unsigned N>
augmentorLib::gaussian_blur_filter_1D< N, true >::gaussian_blur_filter_1D ( ) [delete]
```

7.11.2.2 [gaussian_blur_filter_1D\(\)](#) [2/2]

```
template<unsigned N>
augmentorLib::gaussian_blur_filter_1D< N, true >::gaussian_blur_filter_1D (
    const double sigma ) [inline], [explicit]
```

Definition at line 29 of file filters.h.

7.11.3 Member Function Documentation

7.11.3.1 operator[]()

```
template<unsigned N>
double augmentorLib::gaussian_blur_filter_1D< N, true >::operator[] (
    size_t i ) [inline]
```

Definition at line 47 of file filters.h.

7.11.3.2 size()

```
template<unsigned N>
size_t augmentorLib::gaussian_blur_filter_1D< N, true >::size ( ) [inline]
```

Definition at line 51 of file filters.h.

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

- /Users/gouthamreddykotapalle/Desktop/C++/docu/filters.h

7.12 augmentorLib::GaussianBlurOperation< Image, Kernel > Class Template Reference

```
#include <Operation.h>
```

Inherits [augmentorLib::Operation< Image >](#).

Public Member Functions

- [GaussianBlurOperation](#) (const double sigma, const size_t n, double prob=[UPPER_BOUND_PROB](#), unsigned seed=[NULL_SEED](#))
- [GaussianBlurOperation](#) (const double sigma, double prob=[UPPER_BOUND_PROB](#), unsigned seed=[NULL_SEED](#))
- Image * [perform](#) (Image *image) override

Additional Inherited Members

7.12.1 Detailed Description

```
template<typename Image, int Kernel = 0>
class augmentorLib::GaussianBlurOperation< Image, Kernel >
```

Definition at line 270 of file Operation.h.

7.12.2 Constructor & Destructor Documentation

7.12.2.1 GaussianBlurOperation() [1/2]

```
template<typename Image , int Kernel = 0>
augmentorLib::GaussianBlurOperation< Image, Kernel >::GaussianBlurOperation (
    const double sigma,
    const size_t n,
    double prob = UPPER_BOUND_PROB,
    unsigned seed = NULL_SEED ) [inline], [explicit]
```

Definition at line 274 of file Operation.h.

7.12.2.2 GaussianBlurOperation() [2/2]

```
template<typename Image , int Kernel = 0>
augmentorLib::GaussianBlurOperation< Image, Kernel >::GaussianBlurOperation (
    const double sigma,
    double prob = UPPER_BOUND_PROB,
    unsigned seed = NULL_SEED ) [inline], [explicit]
```

Definition at line 278 of file Operation.h.

7.12.3 Member Function Documentation

7.12.3.1 perform()

```
template<typename Image , int Kernel>
Image * augmentorLib::GaussianBlurOperation< Image, Kernel >::perform (
    Image * image ) [override], [virtual]
```

Perform function that is called to invoke a particular operation

Parameters

<i>image</i>	Image to perform an operation on
--------------	----------------------------------

Returns

A pointer to an image object

Implements [augmentorLib::Operation< Image >](#).

Definition at line 571 of file Operation.h.

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

- /Users/gouthamreddykotapalle/Desktop/C++/docu/[Operation.h](#)

7.13 jpegimageSTL::jpeg::Image Class Reference

```
#include <jpeg.h>
```

Public Types

- typedef uint8_t [pixel_value_type](#)

Public Member Functions

- [Image](#) (size_t x, size_t y, size_t pixelSize=3, int colourSpace=2)
- [Image](#) (const std::string &fileName)
- [Image](#) (const [Image](#) &rhs)
- [~Image](#) ()
Destructor.
- [Image](#) ()
- void [save](#) (const std::string &fileName, int quality=95) const
- size_t [getHeight](#) () const
- size_t [getWidth](#) () const
- size_t [getPixelSize](#) () const
- int [getColorSpace](#) () const
- std::vector< uint8_t > [getPixel](#) (size_t x, size_t y) const
- void [setPixel](#) (size_t x, size_t y, std::vector< uint8_t > pixelValue)
- void [resize](#) (size_t newHeight, size_t newWidth)

7.13.1 Detailed Description

Definition at line 14 of file jpeg.h.

7.13.2 Member Typedef Documentation

7.13.2.1 pixel_value_type

```
typedef uint8_t jpegimageSTL::jpeg::Image::pixel_value_type
```

Definition at line 27 of file jpeg.h.

7.13.3 Constructor & Destructor Documentation

7.13.3.1 Image() [1/4]

```
jpegimageSTL::jpeg::Image::Image (
    size_t x,
    size_t y,
    size_t pixelSize = 3,
    int colourSpace = 2 ) [explicit]
```

[Image](#) constructor

Parameters

<i>x</i>	width of the image
<i>y</i>	height of the image
<i>pixelSize</i>	size of a single pixel
<i>colourSpace</i>	Type of the colourSpace - Currently only supports RGB

See also

[Image\(const std::string& fileName \)](#)

Definition at line 14 of file jpeg.cpp.

7.13.3.2 Image() [2/4]

```
jpegimageSTL::jpeg::Image::Image (
    const std::string & fileName ) [explicit]
```

[Image](#) constructor

Construct with an existing file. Will throw if file cannot be loaded, or is in the wrong format or some other error is encountered.

Parameters

<i>fileName</i>	path to the input file
-----------------	------------------------

Definition at line 56 of file jpeg.cpp.

7.13.3.3 Image() [3/4]

```
jpegimageSTL::jpeg::Image::Image (  
    const Image & rhs )
```

copy constructor

We can construct from an existing image object. This allows us to work on a copy (e.g. shrink then save) without affecting the original we have in memory.

Parameters

<i>rhs</i>	Source image object
------------	---------------------

Definition at line 130 of file jpeg.cpp.

7.13.3.4 ~Image()

```
jpegimageSTL::jpeg::Image::~~Image ( )
```

Destructor.

Definition at line 141 of file jpeg.cpp.

7.13.3.5 Image() [4/4]

```
jpegimageSTL::jpeg::Image::Image ( )
```

Definition at line 42 of file jpeg.cpp.

7.13.4 Member Function Documentation

7.13.4.1 getColorSpace()

```
int jpegimageSTL::jpeg::Image::getColorSpace ( ) const [inline]
```

Definition at line 71 of file jpeg.h.

7.13.4.2 getHeight()

```
size_t jpegimageSTL::jpeg::Image::getHeight ( ) const [inline]
```

Definition at line 68 of file jpeg.h.

7.13.4.3 getPixel()

```
std::vector< uint8_t > jpegimageSTL::jpeg::Image::getPixel (
    size_t x,
    size_t y ) const
```

GetPixel

Will return a vector of pixel components. The vector's size will be 1 for monochrome or 3 for RGB. Elements for the latter will be in order R, G, B.

Parameters

<i>x</i>	x coordinate of the pizel
<i>y</i>	y cooridinate of the pixel

Returns

A pixelvalue: vector of characters that make that pixel

Definition at line 190 of file jpeg.cpp.

7.13.4.4 getPixelSize()

```
size_t jpegimageSTL::jpeg::Image::getPixelSize ( ) const [inline]
```

Definition at line 70 of file jpeg.h.

7.13.4.5 getWidth()

```
size_t jpegimageSTL::jpeg::Image::getWidth ( ) const [inline]
```

Definition at line 69 of file jpeg.h.

7.13.4.6 resize()

```
void jpegimageSTL::jpeg::Image::resize (
    size_t newHeight,
    size_t newWidth )
```

Definition at line 223 of file jpeg.cpp.

7.13.4.7 save()

```
void jpegimageSTL::jpeg::Image::save (
    const std::string & fileName,
    int quality = 95 ) const
```

Save

saves the image to the path specified

Parameters

<i>fileName</i>	output path to save the image to
<i>quality</i>	quality of the image to save

Note

Will throw if file cannot be saved. Quality's usable values are 0-100

Definition at line 145 of file jpeg.cpp.

7.13.4.8 setPixel()

```
void jpegimageSTL::jpeg::Image::setPixel (
    size_t x,
```

```
size_t y,
std::vector< uint8_t > pixelValue )
```

Set Pixel

Used to set one pixel's value at the x,y coordinate. Takes in a vector of subpixels.

Parameters

<i>x</i>	x coordinate of the pizel
<i>y</i>	y coordinate of the pixel
<i>pixelValue</i>	An RGB vector of characters [R, G, B] that make that pixel

Definition at line 207 of file jpeg.cpp.

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

- [/Users/gouthamreddykotapalle/Desktop/C++/docu/jpeg.h](#)
- [/Users/gouthamreddykotapalle/Desktop/C++/docu/jpeg.cpp](#)

7.14 augmentorLib::image_size Struct Reference

```
#include <Operation.h>
```

Public Attributes

- `size_t` [height](#)
- `size_t` [width](#)

7.14.1 Detailed Description

Definition at line 180 of file Operation.h.

7.14.2 Member Data Documentation

7.14.2.1 height

```
size_t augmentorLib::image_size::height
```

Definition at line 181 of file Operation.h.

7.14.2.2 width

```
size_t augmentorLib::image_size::width
```

Definition at line 182 of file Operation.h.

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

- [/Users/gouthamreddykotapalle/Desktop/C++/docu/Operation.h](#)

7.15 augmentorLib::InvertOperation< Image > Class Template Reference

```
#include <Operation.h>
```

Inherits [augmentorLib::Operation< Image >](#).

Public Member Functions

- [InvertOperation](#) (double prob=[UPPER_BOUND_PROB](#), unsigned seed=[NULL_SEED](#))
- Image * [perform](#) (Image *image) override

Additional Inherited Members

7.15.1 Detailed Description

```
template<typename Image>  
class augmentorLib::InvertOperation< Image >
```

Definition at line 260 of file Operation.h.

7.15.2 Constructor & Destructor Documentation

7.15.2.1 InvertOperation()

```
template<typename Image >  
augmentorLib::InvertOperation< Image >::InvertOperation (  
    double prob = UPPER\_BOUND\_PROB,  
    unsigned seed = NULL\_SEED ) [inline], [explicit]
```

Definition at line 262 of file Operation.h.

7.15.3 Member Function Documentation

7.15.3.1 perform()

```
template<typename Image >
Image * augmentorLib::InvertOperation< Image >::perform (
    Image * image ) [override], [virtual]
```

Perform function that is called to invoke a particular operation

Parameters

<i>image</i>	Image to perform an operation on
--------------	----------------------------------

Returns

A pointer to an image object

Implements [augmentorLib::Operation< Image >](#).

Definition at line 545 of file Operation.h.

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

- /Users/gouthamreddykotapalle/Desktop/C++/docu/[Operation.h](#)

7.16 augmentorLib::Operation< Image > Class Template Reference

```
#include <Operation.h>
```

Inherited by [augmentorLib::BoxBlurOperation< Image >](#), [augmentorLib::CropOperation< Image >](#), [augmentorLib::FastGaussianBlurOperation< Image >](#), [augmentorLib::FlipOperation< Image >](#), [augmentorLib::GaussianBlurOperation< Image, Kernel >](#), [augmentorLib::InvertOperation< Image >](#), [augmentorLib::RandomEraseOperation< Image >](#), [augmentorLib::ResizeOperation< Image >](#), [augmentorLib::RotateOperation< Image >](#), [augmentorLib::StdoutOperation< Image >](#), and [augmentorLib::ZoomOperation< Image >](#).

Public Member Functions

- [Operation](#) ()
Default constructor.
- virtual [~Operation](#) ()=default
Destructor for the [Operation](#) class.
- [Operation](#) (double prob, unsigned seed=NULL_SEED)
- template<typename Container >
Container && [perform](#) (Container &&)
- virtual Image * [perform](#) (Image *image)=0

Protected Member Functions

- bool [operate_this_time](#) ()
- [_precision_type](#) [uniform_random_number](#) ()
- [_precision_type](#) [uniform_random_number](#) (const [_precision_type](#) lower, const [_precision_type](#) upper)

7.16.1 Detailed Description

```
template<typename Image>
class augmentorLib::Operation< Image >
```

An operation class that is used is used as a Base class to create other operations

Template Parameters

<i>Image</i>	Takes an Image as a template parameter
--------------	--

Definition at line 119 of file Operation.h.

7.16.2 Constructor & Destructor Documentation

7.16.2.1 Operation() [1/2]

```
template<typename Image >
augmentorLib::Operation< Image >::Operation ( ) [inline]
```

Default constructor.

Definition at line 143 of file Operation.h.

7.16.2.2 ~Operation()

```
template<typename Image >
virtual augmentorLib::Operation< Image >::~~Operation ( ) [virtual], [default]
```

Destructor for the [Operation](#) class.

7.16.2.3 Operation() [2/2]

```
template<typename Image >
augmentorLib::Operation< Image >::Operation (
    double prob,
    unsigned seed = NULL\_SEED ) [inline], [explicit]
```

Parameterized constructor

Parameters

<i>prob</i>	Proability of performing an operation
<i>seed</i>	The random seed for the randomness of the operation

Definition at line 152 of file Operation.h.

7.16.3 Member Function Documentation

7.16.3.1 operate_this_time()

```
template<typename Image >
bool augmentorLib::Operation< Image >::operate_this_time ( ) [inline], [protected]
```

Used to decide whether an operation is performed or not

Returns

A boolean value indicating whether the operation must be performed or not based on probability

Definition at line 128 of file Operation.h.

7.16.3.2 perform() [1/2]

```
template<typename Image >
template<typename Container >
Container && augmentorLib::Operation< Image >::perform (
    Container && container )
```

Definition at line 394 of file Operation.h.

7.16.3.3 perform() [2/2]

```
template<typename Image >
virtual Image* augmentorLib::Operation< Image >::perform (
    Image * image ) [pure virtual]
```

Perform function that is called to invoke a particular operation

Parameters

<i>image</i>	Image to perform an operation on
--------------	----------------------------------

Returns

A pointer to an image object

Implemented in [augmentorLib::FlipOperation< Image >](#), [augmentorLib::RandomEraseOperation< Image >](#), [augmentorLib::FastGaussianBlurOperation< Image >](#), [augmentorLib::BoxBlurOperation< Image >](#), [augmentorLib::GaussianBlurOperation< Image >](#), [augmentorLib::InvertOperation< Image >](#), [augmentorLib::ZoomOperation< Image >](#), [augmentorLib::RotateOperation< Image >](#), [augmentorLib::CropOperation< Image >](#), [augmentorLib::ResizeOperation< Image >](#), and [augmentorLib::StdoutOperation< Image >](#).

7.16.3.4 `uniform_random_number()` [1/2]

```
template<typename Image >
_precision_type augmentorLib::Operation< Image >::uniform_random_number ( ) [inline], [protected]
```

Definition at line 132 of file `Operation.h`.

7.16.3.5 `uniform_random_number()` [2/2]

```
template<typename Image >
_precision_type augmentorLib::Operation< Image >::uniform_random_number (
    const _precision_type lower,
    const _precision_type upper ) [inline], [protected]
```

Definition at line 136 of file `Operation.h`.

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

- `/Users/gouthamreddykotapalle/Desktop/C++/docu/Operation.h`

7.17 `augmentorLib::RandomEraseOperation< Image >` Class Template Reference

```
#include <Operation.h>
```

Inherits [augmentorLib::Operation< Image >](#).

Public Member Functions

- [RandomEraseOperation](#) ([image_size](#) lower_mask_size, [image_size](#) upper_mask_size, double prob=[UPPER_BOUND_PROB](#), unsigned seed=[NULL_SEED](#), unsigned xy_seed=[NULL_SEED](#), unsigned noise_seed=[NULL_SEED](#))
- Image * [perform](#) (Image *image) override

Additional Inherited Members

7.17.1 Detailed Description

```
template<typename Image>
class augmentorLib::RandomEraseOperation< Image >
```

Definition at line 321 of file Operation.h.

7.17.2 Constructor & Destructor Documentation

7.17.2.1 RandomEraseOperation()

```
template<typename Image >
augmentorLib::RandomEraseOperation< Image >::RandomEraseOperation (
    image\_size lower_mask_size,
    image\_size upper_mask_size,
    double prob = UPPER\_BOUND\_PROB,
    unsigned seed = NULL\_SEED,
    unsigned xy_seed = NULL\_SEED,
    unsigned noise_seed = NULL\_SEED ) [inline], [explicit]
```

Definition at line 329 of file Operation.h.

7.17.3 Member Function Documentation

7.17.3.1 perform()

```
template<typename Image >
Image * augmentorLib::RandomEraseOperation< Image >::perform (
    Image * image ) [override], [virtual]
```

Perform function that is called to invoke a particular operation

Parameters

<i>image</i>	Image to perform an operation on
--------------	----------------------------------

Returns

A pointer to an image object

Implements [augmentorLib::Operation< Image >](#).

Definition at line 719 of file Operation.h.

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

- [/Users/gouthamreddykotapalle/Desktop/C++/docu/Operation.h](#)

7.18 augmentorLib::ResizeOperation< Image > Class Template Reference

```
#include <Operation.h>
```

Inherits [augmentorLib::Operation< Image >](#).

Public Member Functions

- [ResizeOperation](#) ()=delete
- [ResizeOperation](#) (image_size lower, image_size upper, double prob=[UPPER_BOUND_PROB](#), unsigned seed=[NULL_SEED](#))
- Image * [perform](#) (Image *image) override

Additional Inherited Members

7.18.1 Detailed Description

```
template<typename Image>  
class augmentorLib::ResizeOperation< Image >
```

Definition at line 186 of file Operation.h.

7.18.2 Constructor & Destructor Documentation

7.18.2.1 ResizeOperation() [1/2]

```
template<typename Image >
augmentorLib::ResizeOperation< Image >::ResizeOperation ( ) [delete]
```

7.18.2.2 ResizeOperation() [2/2]

```
template<typename Image >
augmentorLib::ResizeOperation< Image >::ResizeOperation (
    image_size lower,
    image_size upper,
    double prob = UPPER_BOUND_PROB,
    unsigned seed = NULL_SEED ) [inline], [explicit]
```

Definition at line 194 of file Operation.h.

7.18.3 Member Function Documentation

7.18.3.1 perform()

```
template<typename Image >
Image * augmentorLib::ResizeOperation< Image >::perform (
    Image * image ) [override], [virtual]
```

Perform function that is called to invoke a particular operation

Parameters

<i>image</i>	Image to perform an operation on
--------------	----------------------------------

Returns

A pointer to an image object

Implements [augmentorLib::Operation< Image >](#).

Definition at line 414 of file Operation.h.

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

- [/Users/gouthamreddykotapalle/Desktop/C++/docu/Operation.h](#)

7.19 augmentorLib::rotate_range Struct Reference

```
#include <Operation.h>
```

Public Attributes

- int [min_rotate](#)
- int [max_rotate](#)

7.19.1 Detailed Description

Definition at line 217 of file Operation.h.

7.19.2 Member Data Documentation

7.19.2.1 max_rotate

```
int augmentorLib::rotate_range::max_rotate
```

Definition at line 219 of file Operation.h.

7.19.2.2 min_rotate

```
int augmentorLib::rotate_range::min_rotate
```

Definition at line 218 of file Operation.h.

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

- /Users/gouthamreddykotapalle/Desktop/C++/docu/[Operation.h](#)

7.20 augmentorLib::RotateOperation< Image > Class Template Reference

```
#include <Operation.h>
```

Inherits [augmentorLib::Operation< Image >](#).

Public Member Functions

- [RotateOperation](#) ()=delete
- [RotateOperation](#) ([rotate_range](#) range, double prob=[UPPER_BOUND_PROB](#), unsigned seed=[NULL_SEED](#))
- Image * [perform](#) (Image *image) override

Additional Inherited Members

7.20.1 Detailed Description

```
template<typename Image>
class augmentorLib::RotateOperation< Image >
```

Definition at line 223 of file Operation.h.

7.20.2 Constructor & Destructor Documentation

7.20.2.1 RotateOperation() [1/2]

```
template<typename Image >
augmentorLib::RotateOperation< Image >::RotateOperation ( ) [delete]
```

7.20.2.2 RotateOperation() [2/2]

```
template<typename Image >
augmentorLib::RotateOperation< Image >::RotateOperation (
    rotate_range range,
    double prob = UPPER_BOUND_PROB,
    unsigned seed = NULL_SEED ) [inline], [explicit]
```

Definition at line 230 of file Operation.h.

7.20.3 Member Function Documentation

7.20.3.1 perform()

```
template<typename Image >
Image * augmentorLib::RotateOperation< Image >::perform (
    Image * image ) [override], [virtual]
```

Perform function that is called to invoke a particular operation

Parameters

<i>image</i>	Image to perform an operation on
--------------	----------------------------------

Returns

A pointer to an image object

Implements [augmentorLib::Operation< Image >](#).

Definition at line 504 of file Operation.h.

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

- /Users/gouthamreddykotapalle/Desktop/C++/docu/[Operation.h](#)

7.21 augmentorLib::StdoutOperation< Image > Class Template Reference

```
#include <Operation.h>
```

Inherits [augmentorLib::Operation< Image >](#).

Public Member Functions

- [StdoutOperation](#) ()
- [StdoutOperation](#) (std::string s, double prob=[UPPER_BOUND_PROB](#), unsigned seed=[NULL_SEED](#))
- Image * [perform](#) (Image *image) override

Additional Inherited Members

7.21.1 Detailed Description

```
template<typename Image>  
class augmentorLib::StdoutOperation< Image >
```

Definition at line 167 of file Operation.h.

7.21.2 Constructor & Destructor Documentation

7.21.2.1 StdoutOperation() [1/2]

```
template<typename Image >
augmentorLib::StdoutOperation< Image >::StdoutOperation ( ) [inline]
```

Definition at line 171 of file Operation.h.

7.21.2.2 StdoutOperation() [2/2]

```
template<typename Image >
augmentorLib::StdoutOperation< Image >::StdoutOperation (
    std::string s,
    double prob = UPPER_BOUND_PROB,
    unsigned seed = NULL_SEED ) [inline], [explicit]
```

Definition at line 173 of file Operation.h.

7.21.3 Member Function Documentation

7.21.3.1 perform()

```
template<typename Image >
Image * augmentorLib::StdoutOperation< Image >::perform (
    Image * image ) [override], [virtual]
```

Perform function that is called to invoke a particular operation

Parameters

<i>image</i>	Image to perform an operation on
--------------	----------------------------------

Returns

A pointer to an image object

Implements [augmentorLib::Operation< Image >](#).

Definition at line 404 of file Operation.h.

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

- [/Users/gouthamreddykotapalle/Desktop/C++/docu/Operation.h](#)

7.22 augmentorLib::UniformDistributionGenerator< DataType, IsReal > Class Template Reference

```
#include <Operation.h>
```

7.22.1 Detailed Description

```
template<typename DataType, bool IsReal = std::is_floating_point<DataType>::value>  
class augmentorLib::UniformDistributionGenerator< DataType, IsReal >
```

A class to generate random numbers It has different implementations based on the datatype. It uses `uniform_real_distribution` to generate random numbers of floating numbers, and `uniform_int_distribution` to generate integers.

Definition at line 33 of file `Operation.h`.

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

- [/Users/gouthamreddykotapalle/Desktop/C++/docu/Operation.h](#)

7.23 augmentorLib::UniformDistributionGenerator< DataType, false > Class Template Reference

```
#include <Operation.h>
```

Public Member Functions

- [UniformDistributionGenerator](#) ()
- [~UniformDistributionGenerator](#) ()=default
- [UniformDistributionGenerator](#) (unsigned seed)
- [UniformDistributionGenerator](#) (unsigned seed, DataType lower, DataType upper)
- DataType [operator](#)() ()

7.23.1 Detailed Description

```
template<typename DataType>  
class augmentorLib::UniformDistributionGenerator< DataType, false >
```

A class to generate int numbers Int numbers include from int8 to unsigned long long.

Definition at line 77 of file `Operation.h`.

7.23.2 Constructor & Destructor Documentation

7.23.2.1 UniformDistributionGenerator() [1/3]

```
template<typename DataType >
augmentorLib::UniformDistributionGenerator< DataType, false >::UniformDistributionGenerator ( )
[inline]
```

Definition at line 90 of file Operation.h.

7.23.2.2 ~UniformDistributionGenerator()

```
template<typename DataType >
augmentorLib::UniformDistributionGenerator< DataType, false >::~~UniformDistributionGenerator ( )
[default]
```

7.23.2.3 UniformDistributionGenerator() [2/3]

```
template<typename DataType >
augmentorLib::UniformDistributionGenerator< DataType, false >::UniformDistributionGenerator (
    unsigned seed ) [inline], [explicit]
```

if seed = 0, the program will automatically generate a seed based on current time. The range of random numbers from Datatype_min to Datatype_max

Definition at line 98 of file Operation.h.

7.23.2.4 UniformDistributionGenerator() [3/3]

```
template<typename DataType >
augmentorLib::UniformDistributionGenerator< DataType, false >::UniformDistributionGenerator (
    unsigned seed,
    DataType lower,
    DataType upper ) [inline], [explicit]
```

if seed = 0, the program will automatically generate a seed based on current time.

Definition at line 105 of file Operation.h.

7.23.3 Member Function Documentation

7.23.3.1 operator>()

```
template<typename DataType >  
DataType augmentorLib::UniformDistributionGenerator< DataType, false >::operator() ( ) [inline]
```

Definition at line 109 of file Operation.h.

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

- /Users/gouthamreddykotapalle/Desktop/C++/docu/[Operation.h](#)

7.24 augmentorLib::UniformDistributionGenerator< DataType, true > Class Template Reference

```
#include <Operation.h>
```

Public Member Functions

- [UniformDistributionGenerator](#) ()
- [~UniformDistributionGenerator](#) ()=default
- [UniformDistributionGenerator](#) (unsigned seed)
- [UniformDistributionGenerator](#) (unsigned seed, DataType lower, DataType upper)
- DataType [operator\(\)](#) ()

7.24.1 Detailed Description

```
template<typename DataType>  
class augmentorLib::UniformDistributionGenerator< DataType, true >
```

A class to generate floating numbers

Definition at line 39 of file Operation.h.

7.24.2 Constructor & Destructor Documentation

7.24.2.1 UniformDistributionGenerator() [1/3]

```
template<typename DataType >
augmentorLib::UniformDistributionGenerator< DataType, true >::UniformDistributionGenerator ( )
[inline]
```

Definition at line 52 of file Operation.h.

7.24.2.2 ~UniformDistributionGenerator()

```
template<typename DataType >
augmentorLib::UniformDistributionGenerator< DataType, true >::~~UniformDistributionGenerator ( )
[default]
```

7.24.2.3 UniformDistributionGenerator() [2/3]

```
template<typename DataType >
augmentorLib::UniformDistributionGenerator< DataType, true >::UniformDistributionGenerator (
    unsigned seed ) [inline], [explicit]
```

if seed = 0, the program will automatically generate a seed based on current time. The range of random numbers is from 0 to 1.

Definition at line 60 of file Operation.h.

7.24.2.4 UniformDistributionGenerator() [3/3]

```
template<typename DataType >
augmentorLib::UniformDistributionGenerator< DataType, true >::UniformDistributionGenerator (
    unsigned seed,
    DataType lower,
    DataType upper ) [inline], [explicit]
```

Definition at line 64 of file Operation.h.

7.24.3 Member Function Documentation

7.24.3.1 operator()()

```
template<typename DataType >  
DataType augmentorLib::UniformDistributionGenerator< DataType, true >::operator() ( ) [inline]
```

Definition at line 68 of file Operation.h.

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

- /Users/gouthamreddykotapalle/Desktop/C++/docu/[Operation.h](#)

7.25 augmentorLib::zoom_factor Struct Reference

```
#include <Operation.h>
```

Public Attributes

- double [min_factor](#)
- double [max_factor](#)

7.25.1 Detailed Description

Definition at line 237 of file Operation.h.

7.25.2 Member Data Documentation

7.25.2.1 max_factor

```
double augmentorLib::zoom_factor::max_factor
```

Definition at line 239 of file Operation.h.

7.25.2.2 min_factor

```
double augmentorLib::zoom_factor::min_factor
```

Definition at line 238 of file Operation.h.

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

- /Users/gouthamreddykotapalle/Desktop/C++/docu/[Operation.h](#)

7.26 augmentorLib::ZoomOperation< Image > Class Template Reference

```
#include <Operation.h>
```

Inherits [augmentorLib::Operation< Image >](#).

Public Member Functions

- [ZoomOperation](#) ()=delete
- [ZoomOperation](#) ([zoom_factor](#) factor, double prob=[UPPER_BOUND_PROB](#), unsigned seed=[NULL_SEED](#))
- Image * [perform](#) (Image *image) override

Additional Inherited Members

7.26.1 Detailed Description

```
template<typename Image>  
class augmentorLib::ZoomOperation< Image >
```

Definition at line 243 of file Operation.h.

7.26.2 Constructor & Destructor Documentation

7.26.2.1 ZoomOperation() [1/2]

```
template<typename Image >  
augmentorLib::ZoomOperation< Image >::ZoomOperation ( ) [delete]
```

7.26.2.2 ZoomOperation() [2/2]

```
template<typename Image >  
augmentorLib::ZoomOperation< Image >::ZoomOperation (   
    zoom\_factor factor,  
    double prob = UPPER\_BOUND\_PROB,  
    unsigned seed = NULL\_SEED ) [inline], [explicit]
```

Definition at line 251 of file Operation.h.

7.26.3 Member Function Documentation

7.26.3.1 perform()

```
template<typename Image >  
Image * augmentorLib::ZoomOperation< Image >::perform (  
    Image * image )    [override], [virtual]
```

Perform function that is called to invoke a particular operation

Parameters

<i>image</i>	Image to perform an operation on
--------------	----------------------------------

Returns

A pointer to an image object

Implements [augmentorLib::Operation< Image >](#).

Definition at line 477 of file Operation.h.

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

- [/Users/gouthamreddykotapalle/Desktop/C++/docu/Operation.h](#)

Chapter 8

File Documentation

8.1 /Users/gouthamreddykotapalle/Desktop/C++/docu/Augmentor Design Document.md File Reference

8.2 /Users/gouthamreddykotapalle/Desktop/C++/docu/Augmentor.cpp File Reference

```
#include "Augmentor.h"  
#include <filesystem>  
#include <chrono>
```

Namespaces

- [augmentorLib](#)

Typedefs

- typedef std::chrono::high_resolution_clock [clocking](#)

8.2.1 Typedef Documentation

8.2.1.1 clocking

```
typedef std::chrono::high_resolution_clock clocking
```

Definition at line 5 of file Augmentor.cpp.

8.3 /Users/gouthamreddykotapalle/Desktop/C++/docu/Augmentor.h File Reference

```
#include "jpeg.h"
#include "Operation.h"
#include <iostream>
#include <string>
#include <stdexcept>
#include <vector>
#include <memory>
```

Classes

- class [augmentorLib::Augmentor](#)

Namespaces

- [augmentorLib](#)

8.4 /Users/gouthamreddykotapalle/Desktop/C++/docu/CMakeLists.txt File Reference

8.5 /Users/gouthamreddykotapalle/Desktop/C++/docu/filters.h File Reference

```
#include <vector>
#include <cmath>
#include <numeric>
```

Classes

- class [augmentorLib::gaussian_blur_filter_1D< N, Static >](#)
- class [augmentorLib::gaussian_blur_filter_1D< N, true >](#)
- class [augmentorLib::gaussian_blur_filter_1D< N, false >](#)
- struct [augmentorLib::box_blur_filter_1D](#)

Namespaces

- [augmentorLib](#)

8.6 /Users/gouthamreddykotapalle/Desktop/C++/docu/jpeg.cpp File Reference

```
#include "jpeg.h"
#include <jpeglib.h>
#include <fstream>
#include <stdexcept>
#include <string>
#include <vector>
#include <iostream>
```

Namespaces

- [jpegimageSTL](#)
- [jpegimageSTL::jpeg](#)

8.7 /Users/gouthamreddykotapalle/Desktop/C++/docu/jpeg.h File Reference

```
#include <cstdint>
#include <memory>
#include <string>
#include <vector>
```

Classes

- class [jpegimageSTL::jpeg::Image](#)

Namespaces

- [jpegimageSTL](#)
- [jpegimageSTL::jpeg](#)

8.8 /Users/gouthamreddykotapalle/Desktop/C++/docu/main.cpp File Reference

```
#include <iostream>
#include <vector>
#include "Augmentor.h"
#include "Operation.h"
```

Typedefs

- typedef chrono::high_resolution_clock [clocking](#)

Functions

- int [main](#) (int argc, char *argv[])

8.8.1 Typedef Documentation

8.8.1.1 clocking

```
typedef chrono::high_resolution_clock clocking
```

Definition at line 9 of file main.cpp.

8.8.2 Function Documentation

8.8.2.1 main()

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

Definition at line 11 of file main.cpp.

8.9 /Users/gouthamreddykotapalle/Desktop/C++/docu/Operation.cpp File Reference

```
#include "Operation.h"
```

8.10 /Users/gouthamreddykotapalle/Desktop/C++/docu/Operation.h File Reference

```
#include <random>
#include <chrono>
#include <utility>
#include <iostream>
#include "filters.h"
#include <algorithm>
#include <vector>
#include <limits>
#include <type_traits>
```

Classes

- class [augmentorLib::UniformDistributionGenerator< DataType, IsReal >](#)
- class [augmentorLib::UniformDistributionGenerator< DataType, true >](#)
- class [augmentorLib::UniformDistributionGenerator< DataType, false >](#)
- class [augmentorLib::Operation< Image >](#)
- class [augmentorLib::StdoutOperation< Image >](#)
- struct [augmentorLib::image_size](#)
- class [augmentorLib::ResizeOperation< Image >](#)
- class [augmentorLib::CropOperation< Image >](#)
- struct [augmentorLib::rotate_range](#)
- class [augmentorLib::RotateOperation< Image >](#)
- struct [augmentorLib::zoom_factor](#)
- class [augmentorLib::ZoomOperation< Image >](#)
- class [augmentorLib::InvertOperation< Image >](#)
- class [augmentorLib::GaussianBlurOperation< Image, Kernel >](#)
- class [augmentorLib::BoxBlurOperation< Image >](#)
- class [augmentorLib::FastGaussianBlurOperation< Image >](#)
- class [augmentorLib::RandomEraseOperation< Image >](#)
- class [augmentorLib::FlipOperation< Image >](#)
- struct [augmentorLib::accumulator](#)

Namespaces

- [augmentorLib](#)

Macros

- [#define PI 3.14159](#)
- [#define HORIZONTAL "Horizontal"](#)
- [#define VERTICAL "Vertical"](#)

Functions

- void [augmentorLib::convert2pixel](#) (const std::vector< double > &src, std::vector< uint8_t > &target, size_t n)

Variables

- const double [augmentorLib::LOWER_BOUND_PROB](#) = 0.0
- const double [augmentorLib::UPPER_BOUND_PROB](#) = 1.0
- const unsigned [augmentorLib::NULL_SEED](#) = 0

8.10.1 Macro Definition Documentation

8.10.1.1 HORIZONTAL

```
#define HORIZONTAL "Horizontal"
```

Definition at line 9 of file Operation.h.

8.10.1.2 PI

```
#define PI 3.14159
```

Definition at line 8 of file Operation.h.

8.10.1.3 VERTICAL

```
#define VERTICAL "Vertical"
```

Definition at line 10 of file Operation.h.

8.11 /Users/gouthamreddykotapalle/Desktop/C++/docu/unit_test.cpp File Reference

```
#include "gtest/gtest.h"
#include "Augmentor.h"
#include "jpeg.h"
```

Classes

- class [AugmentorTest](#)

Functions

- [TEST_F](#) ([AugmentorTest](#), [resize0](#))
- [TEST_F](#) ([AugmentorTest](#), [invert0](#))
- [TEST_F](#) ([AugmentorTest](#), [zoom0](#))
- [TEST_F](#) ([AugmentorTest](#), [crop0](#))
- [TEST_F](#) ([AugmentorTest](#), [rotate0](#))
- int [main](#) (int argc, char **argv)

8.11.1 Function Documentation

8.11.1.1 main()

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

Definition at line 92 of file unit_test.cpp.

8.11.1.2 TEST_F() [1/5]

```
TEST_F (  
    AugmentorTest ,  
    crop0 )
```

Definition at line 68 of file unit_test.cpp.

8.11.1.3 TEST_F() [2/5]

```
TEST_F (  
    AugmentorTest ,  
    invert0 )
```

Definition at line 46 of file unit_test.cpp.

8.11.1.4 TEST_F() [3/5]

```
TEST_F (
    AugmentorTest ,
    resize0 )
```

Definition at line 35 of file unit_test.cpp.

8.11.1.5 TEST_F() [4/5]

```
TEST_F (
    AugmentorTest ,
    rotate0 )
```

Definition at line 79 of file unit_test.cpp.

8.11.1.6 TEST_F() [5/5]

```
TEST_F (
    AugmentorTest ,
    zoom0 )
```

Definition at line 59 of file unit_test.cpp.

Index

`/Users/gouthamreddykotapalle/Desktop/C++/docu/Augmentor.cpp`
 Design Document.md, 68
`/Users/gouthamreddykotapalle/Desktop/C++/docu/AugmentorLib.cpp`
 68
`/Users/gouthamreddykotapalle/Desktop/C++/docu/Augmentor.h`
 69
`/Users/gouthamreddykotapalle/Desktop/C++/docu/CMakeLists.txt`
 69
`/Users/gouthamreddykotapalle/Desktop/C++/docu/Operation.cpp`
 71
`/Users/gouthamreddykotapalle/Desktop/C++/docu/Operation.h`
 72
`/Users/gouthamreddykotapalle/Desktop/C++/docu/filters.h`
 69
`/Users/gouthamreddykotapalle/Desktop/C++/docu/jpeg.cpp`
 70
`/Users/gouthamreddykotapalle/Desktop/C++/docu/jpeg.h`
 70
`/Users/gouthamreddykotapalle/Desktop/C++/docu/main.cpp`
 70
`/Users/gouthamreddykotapalle/Desktop/C++/docu/unit_test.cpp`
 73
 _datatype
 augmentorLib::accumulator, 14
 ~Augmentor
 augmentorLib::Augmentor, 16
 ~AugmentorTest
 AugmentorTest, 27
 ~Image
 jpegimageSTL::jpeg::Image, 44
 ~Operation
 augmentorLib::Operation< Image >, 50
 ~UniformDistributionGenerator
 augmentorLib::UniformDistributionGenerator<
 DataType, false >, 61
 augmentorLib::UniformDistributionGenerator<
 DataType, true >, 63

accumulator
 augmentorLib::accumulator, 14
add
 augmentorLib::accumulator, 14
aug
 AugmentorTest, 28
Augmentor
 augmentorLib::Augmentor, 16
 Augmentor.cpp
 clocking, 68
 augmentorLib, 10
 convert2pixel, 11
 LOWER_BOUND_PROB, 11
 NULL_SEED, 11
 UPPER_BOUND_PROB, 11
 augmentorLib::accumulator, 13
 _datatype, 14
 accumulator, 14
 add, 14
 div, 14
 shift, 14
 values, 15
 augmentorLib::Augmentor, 15
 ~Augmentor, 16
 Augmentor, 16
 blur, 17
 crop, 18
 flip, 18
 invert, 20
 pipeline, 20
 random_erase, 20, 21
 rapid_blur, 22
 resize, 22–24
 rotate, 24
 sample, 25
 save, 25
 zoom, 26
 augmentorLib::box_blur_filter_1D, 28
 box_blur_filter_1D, 29
 length, 30
 pseudo_gaussian_filter, 29
 augmentorLib::BoxBlurOperation< Image >, 30
 BoxBlurOperation, 31
 perform, 31
 augmentorLib::CropOperation< Image >, 32
 CropOperation, 32
 perform, 33
 augmentorLib::FastGaussianBlurOperation< Image >, 33
 FastGaussianBlurOperation, 34
 perform, 34
 augmentorLib::FlipOperation< Image >, 35
 FlipOperation, 35
 perform, 36

- augmentorLib::gaussian_blur_filter_1D< N, false >, 37
 - gaussian_blur_filter_1D, 37, 38
 - operator[], 38
 - size, 38
- augmentorLib::gaussian_blur_filter_1D< N, Static >, 36
- augmentorLib::gaussian_blur_filter_1D< N, true >, 38
 - gaussian_blur_filter_1D, 39
 - operator[], 39
 - size, 40
- augmentorLib::GaussianBlurOperation< Image, Kernel >, 40
 - GaussianBlurOperation, 41
 - perform, 41
- augmentorLib::image_size, 47
 - height, 47
 - width, 47
- augmentorLib::InvertOperation< Image >, 48
 - InvertOperation, 48
 - perform, 49
- augmentorLib::Operation< Image >, 49
 - ~Operation, 50
 - operate_this_time, 51
 - Operation, 50
 - perform, 51
 - uniform_random_number, 52
- augmentorLib::RandomEraseOperation< Image >, 52
 - perform, 53
 - RandomEraseOperation, 53
- augmentorLib::ResizeOperation< Image >, 54
 - perform, 55
 - ResizeOperation, 54, 55
- augmentorLib::rotate_range, 56
 - max_rotate, 56
 - min_rotate, 56
- augmentorLib::RotateOperation< Image >, 56
 - perform, 57
 - RotateOperation, 57
- augmentorLib::StdoutOperation< Image >, 58
 - perform, 59
 - StdoutOperation, 58, 59
- augmentorLib::UniformDistributionGenerator< DataType, false >, 60
 - ~UniformDistributionGenerator, 61
 - operator(), 62
 - UniformDistributionGenerator, 61
- augmentorLib::UniformDistributionGenerator< DataType, IsReal >, 60
- augmentorLib::UniformDistributionGenerator< DataType, true >, 62
 - ~UniformDistributionGenerator, 63
 - operator(), 63
 - UniformDistributionGenerator, 62, 63
- augmentorLib::zoom_factor, 64
 - max_factor, 64
 - min_factor, 64
- augmentorLib::ZoomOperation< Image >, 65
 - perform, 66
 - ZoomOperation, 65
- AugmentorTest, 26
 - ~AugmentorTest, 27
 - aug, 28
 - AugmentorTest, 27
 - img, 28
 - SetUp, 27
 - TearDown, 27
- blur
 - augmentorLib::Augmentor, 17
- box_blur_filter_1D
 - augmentorLib::box_blur_filter_1D, 29
- BoxBlurOperation
 - augmentorLib::BoxBlurOperation< Image >, 31
- clocking
 - Augmentor.cpp, 68
 - main.cpp, 71
- convert2pixel
 - augmentorLib, 11
- crop
 - augmentorLib::Augmentor, 18
- CropOperation
 - augmentorLib::CropOperation< Image >, 32
- div
 - augmentorLib::accumulator, 14
- FastGaussianBlurOperation
 - augmentorLib::FastGaussianBlurOperation< Image >, 34
- flip
 - augmentorLib::Augmentor, 18
- FlipOperation
 - augmentorLib::FlipOperation< Image >, 35
- gaussian_blur_filter_1D
 - augmentorLib::gaussian_blur_filter_1D< N, false >, 37, 38
 - augmentorLib::gaussian_blur_filter_1D< N, true >, 39
- GaussianBlurOperation
 - augmentorLib::GaussianBlurOperation< Image, Kernel >, 41
- getColorSpace
 - jpegimageSTL::jpeg::Image, 44
- getHeight
 - jpegimageSTL::jpeg::Image, 45
- getPixel
 - jpegimageSTL::jpeg::Image, 45
- getPixelSize

- jpegimageSTL::jpeg::Image, 45
- getWidth
 - jpegimageSTL::jpeg::Image, 45
- height
 - augmentorLib::image_size, 47
- HORIZONTAL
 - Operation.h, 73
- Image
 - jpegimageSTL::jpeg::Image, 43, 44
- img
 - AugmentorTest, 28
- invert
 - augmentorLib::Augmentor, 20
- InvertOperation
 - augmentorLib::InvertOperation< Image >, 48
- jpegimageSTL, 12
- jpegimageSTL::jpeg, 12
- jpegimageSTL::jpeg::Image, 42
 - ~Image, 44
 - getColorSpace, 44
 - getHeight, 45
 - getPixel, 45
 - getPixelSize, 45
 - getWidth, 45
 - Image, 43, 44
 - pixel_value_type, 42
 - resize, 46
 - save, 46
 - setPixel, 46
- length
 - augmentorLib::box_blur_filter_1D, 30
- LOWER_BOUND_PROB
 - augmentorLib, 11
- main
 - main.cpp, 71
 - unit_test.cpp, 74
- main.cpp
 - clocking, 71
 - main, 71
- max_factor
 - augmentorLib::zoom_factor, 64
- max_rotate
 - augmentorLib::rotate_range, 56
- min_factor
 - augmentorLib::zoom_factor, 64
- min_rotate
 - augmentorLib::rotate_range, 56
- NULL_SEED
 - augmentorLib, 11
- operate_this_time
 - augmentorLib::Operation< Image >, 51
- Operation
 - augmentorLib::Operation< Image >, 50
- Operation.h
 - HORIZONTAL, 73
 - PI, 73
 - VERTICAL, 73
- operator()
 - augmentorLib::UniformDistributionGenerator<
 - DataType, false >, 62
 - augmentorLib::UniformDistributionGenerator<
 - DataType, true >, 63
- operator[]
 - augmentorLib::gaussian_blur_filter_1D< N, false >, 38
 - augmentorLib::gaussian_blur_filter_1D< N, true >, 39
- perform
 - augmentorLib::BoxBlurOperation< Image >, 31
 - augmentorLib::CropOperation< Image >, 33
 - augmentorLib::FastGaussianBlurOperation< Image
 - >, 34
 - augmentorLib::FlipOperation< Image >, 36
 - augmentorLib::GaussianBlurOperation< Image,
 - Kernel >, 41
 - augmentorLib::InvertOperation< Image >, 49
 - augmentorLib::Operation< Image >, 51
 - augmentorLib::RandomEraseOperation< Image >, 53
 - augmentorLib::ResizeOperation< Image >, 55
 - augmentorLib::RotateOperation< Image >, 57
 - augmentorLib::StdoutOperation< Image >, 59
 - augmentorLib::ZoomOperation< Image >, 66
- PI
 - Operation.h, 73
- pipeline
 - augmentorLib::Augmentor, 20
- pixel_value_type
 - jpegimageSTL::jpeg::Image, 42
- pseudo_gaussian_filter
 - augmentorLib::box_blur_filter_1D, 29
- random_erase
 - augmentorLib::Augmentor, 20, 21
- RandomEraseOperation
 - augmentorLib::RandomEraseOperation< Image >, 53
- rapid_blur
 - augmentorLib::Augmentor, 22
- resize
 - augmentorLib::Augmentor, 22–24
 - jpegimageSTL::jpeg::Image, 46
- ResizeOperation

- augmentorLib::ResizeOperation< Image >, [54](#), [55](#)
- rotate
 - augmentorLib::Augmentor, [24](#)
- RotateOperation
 - augmentorLib::RotateOperation< Image >, [57](#)
- sample
 - augmentorLib::Augmentor, [25](#)
- save
 - augmentorLib::Augmentor, [25](#)
 - jpegimageSTL::jpeg::Image, [46](#)
- setPixel
 - jpegimageSTL::jpeg::Image, [46](#)
- SetUp
 - AugmentorTest, [27](#)
- shift
 - augmentorLib::accumulator, [14](#)
- size
 - augmentorLib::gaussian_blur_filter_1D< N, false >, [38](#)
 - augmentorLib::gaussian_blur_filter_1D< N, true >, [40](#)
- StdoutOperation
 - augmentorLib::StdoutOperation< Image >, [58](#), [59](#)
- TearDown
 - AugmentorTest, [27](#)
- TEST_F
 - unit_test.cpp, [74](#), [75](#)
- uniform_random_number
 - augmentorLib::Operation< Image >, [52](#)
- UniformDistributionGenerator
 - augmentorLib::UniformDistributionGenerator< DataType, false >, [61](#)
 - augmentorLib::UniformDistributionGenerator< DataType, true >, [62](#), [63](#)
- unit_test.cpp
 - main, [74](#)
 - TEST_F, [74](#), [75](#)
- UPPER_BOUND_PROB
 - augmentorLib, [11](#)
- values
 - augmentorLib::accumulator, [15](#)
- VERTICAL
 - Operation.h, [73](#)
- width
 - augmentorLib::image_size, [47](#)
- zoom
 - augmentorLib::Augmentor, [26](#)
- ZoomOperation
 - augmentorLib::ZoomOperation< Image >, [65](#)