Open Source Face Image Quality (OFIQ)

Chapter 1

Open Source Face Image Quality (OFIQ) Library

1.1 Introduction

OFIQ (Open Source Face Image Quality) is a software library for computing quality aspects of a facial image. OFIQ is written in the C/C++ programming language. OFIQ is the reference implementation for the ISO/IEC 29794-5 international standard; see https://bsi.bund.de/dok/OFIQ-e.

1.2 License

This is the source code of OFIQ. OFIQ is a software for assessing the quality of facial image properties and potential defects. OFIQ is licensed under the MIT licenses (see text below). It includes dependencies that may be licensed otherwise. A documentation on the license situation of dependencies can be found in the table below.

1.2.1 OFIQ License

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2.2 License of dependencies

In the table the license situation of the files shipped with the OFIQ source is documented.

File/directory	Description	Reference	Original license
CMakeLists.txt	CMake file for building OFIQ	-	OFIQ license
README.md	Readme file	-	OFIQ license
LICENSE.md	File containing license information	-	OFIQ license
Version.txt	Contains OFIQ's current version information.	-	OFIQ license
cmake/	Folder containing cmake helper scripts	https://cmake.↩org/	OFIQ license
data/ofiq_config.jaxn	OFIQ's configuration file	-	OFIQ license
data/models/	Folder containing a single readme file; to this folder, model files are downloaded from the ISO portal when running one of the building scripts. Note, the readme file is subject to OFIQ's license. The license situation of the model files is documented separately in license files placed in the sub-directories after download.	-	-
data/tests/expected_↩results/expected_results.↩csv	CSV file with expected native quality scores and quality component values used for running conformance tests.	-	OFIQ license
data/tests/images/	Folder containing a single readme file; to this folder, conformance test images are downloaded from the ISO portal when running one of the building scripts. Note, the readme file is subject to OFIQ's license. The license situation of the image files is documented separately in a license file downloaded with the images.	-	-
conan/	Directory containing files that are used for the Conan package manager.	https://conan.io/	OFIQ license
OFIQlib/	Directory containing the OFIQ source code (including headers).	-	OFIQ license
doc/src/mainpage.h	Doxygen documentation of OFIQ	-	OFIQ license

File/directory	Description	Reference	Original license
doc/src/ofiq-doxygen.cfg	Configuration for building doxygen documentation. Has been generated by doxygen and was edited afterwards. May be affected by copyleft. Was used to generate doc/refman.pdf which does not, however, need to be linked with any software compilation using OFIQ.	https://www.↩ doxygen.org/	mixed
doc/refman.pdf	Documentation of OFIQ. Generated by doxygen and LaTeX. The documentation does not need to be linked with any software compilation using OFIQ.	https://www.↩ doxygen.org/, https://www.↩ latex-project.org/	mixed
scripts/build.cmd	Script for building OFIQ-↩ Release on Windows; includes steps for installing packages via conan and downloading models and conformance test images from ISO portal.	-	OFIQ license
scripts/build_debug.cmd	Script for building OFIQ-↩ Debug on Windows; includes steps for installing packages via conan and downloading models and conformance test images from ISO portal.	-	OFIQ license
scripts/build.sh	Script for building OFIQ-↩ Release on Linux; includes steps for installing packages via conan and downloading models and conformance test images from ISO portal.	-	OFIQ license
scripts/build_debug.sh	Script for building OFIQ-↩ Debug on Linux; includes steps for installing packages via conan and downloading models and conformance test images from ISO portal.	-	OFIQ license
scripts/conformance_↩ tests.cmd	Runs conformance tests with OFIQ-Release on Windows	-	OFIQ license
scripts/conformance_↩ tests_debug.cmd	Runs conformance tests with OFIQ-Debug on Windows	-	OFIQ license
scripts/conformance_↩ tests.sh	Runs conformance tests with OFIQ-Release on Linux	-	OFIQ license
scripts/conformance_↩ tests_debug.sh	Runs conformance tests with OFIQ-Debug on Linux	-	OFIQ license

File/directory	Description	Reference	Original license
testing/	Directory containing OFIQ source code running confor- mance tests	-	OFIQ license

1.3 Compilation

To build OFIQ you need to install Python with pip, cmake and conan.

- Python (version 3.10.12 or higher)

- Download and install cmake (version 3.26 or higher)

- Download and install conan (version 2.0.17)

1.3.1 Linux

The following has been tested on an Ubuntu 22.04 (64 bit) installation.
Install necessary packages.

```
$ sudo apt-get install build-essential python3-pip liblapack-dev
```

To install cmake (in a version 3.26 or higher) use snap (and not apt) as follows.

```
$ sudo snap install cmake --classic
```

Conan is installed via Python with

```
pip install conan==2.0.17
```

In order to build OFIQ and installing all required packages via conan run the following.

```
$ cd <OFIQ-SOURCE>/scripts
$ sh build.sh
```

where <OFIQ-SOURCE> denotes the root folder of the OFIQ source files.
This will create the following output.

file/directory	description
build/	Folder with the Linux build including the binaries `OFIQSampleApp` and `libofiq←_lib.so`.
build/conan/	Conan cache with packages downloaded.
install_x86_64_linux/	Folder with the installation including the binaries `Release/bin/OFIQSampleApp`, `Release/lib/libofiq_lib.so`, `Release/lib/libonnxruntime.←so.1.14.1`, and the header files in `Release/include/`.
data/models/	Model files downloaded from the ISO portal during build process (see Download model files).
data/tests/images/	Conformance test images downloaded from the ISO portal (see Download conformance test images).

1.3.2 Windows

The following has been tested on a Windows 10 (64 bit) installation using a **Python installation version 3.11.5** with pip package such that the `pip` command can be executed from the command prompt. Furthermore, an installation of **cmake version 3.29** has been used. As the compiler, **Microsoft's Visual Studio 2019** was used.

To install conan, run

```
$ pip install conan==2.0.17
```

from the command prompt.

In order to build OFIQ and install all required packages run the following.

```
$ cd <OFIQ-SOURCE>\scripts\
$ .\build.cmd
```

where <OFIQ-SOURCE> denotes the root folder of the OFIQ source files.
This will create the following output.

file/directory	description
build\build_win\	Folder with the Visual Studio solution files placed and pre-compilation.
build\conan\	Conan cache with packages downloaded.
install_x86_64\	Folder with the OFIQ installation files. This includes the binaries `Release\bin\OFIQSampleApp`, `Release\bin\libofiq_lib.so`, `Release\bin\libonnxruntime.so.1.14.1`, and the header files in `Release\include\`.
data\models\	Model files downloaded from the ISO portal during build process (see Download model files).
data\tests\images\	Conformance test images downloaded from the ISO portal (see Download conformance test images).

1.3.3 Download model files

To run OFIQ, the model files from the ISO portal need to be downloaded and be placed in the `./data/models/` directory. This step is integrated in the cmake building process.

1.3.4 Download conformance test images

To run conformance tests, the conformance test images need to be downloaded from the ISO portal and be placed in the `./data/tests/images/` directory. This step is integrated in the cmake building process.

1.4 Running conformance tests

The conformance tests are executed by going to `<OFIQ-SOURCE>/scripts/` and run `conformance_↩tests.cmd` (Windows) or `conformance_tests.sh` (Linux).

1.5 Running the sample executable

In this section, we describe how to run the sample application of OFIQ after compilation (see Compilation). A documentation for the arguments that can be passed can be found below.

1.5.1 Quality assessment for a single facial image

The sample application takes an images and outputs the computed quality assessments. For example, to output the quality assessments for one of the conformance test images using OFIQ's configuration (in `./data`), run the following commands on Linux.

```
$ cd <OFIQ-SOURCE>/install_x86_64_linux/Release/bin/
$ ./OFIQSampleApp -c ../../../data/ofiq_config.jaxn -i
        ../../../data/tests/images/b-01-smile.png
```

On Windows run the following commands.

```
$ cd <OFIQ-SOURCE>\install_x86_64\Release\bin
$ .\OFIQSampleApp -c ..\..\..\data\ofiq_config.jaxn -i
        ..\..\..\data\tests\images\b-01-smile.png
```

1.5.2 Quality assessment for multiple images

To reproduce the conformance test table given in Annex A of the ISO/IEC 29794-5 international standard for all conformance test images, run the following commands on Linux.

```
$ cd <OFIQ-SOURCE>/install_x86_64_linux/Release/bin/
$ ./OFIQSampleApp -c ..\..\..\data\ofiq_config.jaxn -i
        ..\..\..\data\tests\images\ -o table.csv
```

The result will be written in the file `<OFIQ-SOURCE>/install_x86_64_linux/Release/bin/table.csv`.

On Windows run the following commands.

```
$ cd <OFIQ-SOURCE>\install_x86_64\Release\bin\
$ .\OFIQSampleApp -c ..\..\..\data\ -i
        ..\..\..\data\tests\images\ -o table.csv
```

The result will be written in the file `<OFIQ-SOURCE>\install_x86_64\Release\bin\table.csv`.

1.5.3 Arguments

The usage pattern of the sample application is the following.

```
OFIQSampleApp
  -c <directory or file path>
  [-cf <config file name>]
  -i <directory or image file path>
  [-o <csv file path>]
```

The following table documents the usage of the sample application.

flag	argument
-c	Path to a directory containing the file ofiq_config.jaxn or a path to a JAXN configuration file (see Configuration).
-cf	Name of the JAXN configuration file contained in the directory specified by the flag -c. Must be omitted if -c specifies a path to a file.
-i	Path to a directory containing facial images or a path to a facial image file. If a directory path is specified, all images in PNG and JPEG format will be processed.
-o	Path to a CSV file to where the quality assessment is written. If -o is not specified, the output is written to the standard output.

1.6 Configuration

In this section, we describe the configuration file for OFIQ. OFIQ uses a JSON-like configuration based on the `taoJSON` library. Using taoJSON, OFIQ reads its configuration from a JAXN-formatted file.

A minimal configuration working with OFIQ looks as follows.

```
{
 "config": {
  "detector": "ssd",
  "landmarks": "ADNet",
  "measures": [
   "UnifiedQualityScore"
  ],
  "params": {
   "detector": {
    "ssd": {
     "model_path": "models/face_detection/ssd_facedetect.caffemodel",
     "prototxt_path": "models/face_detection/ssd_facedetect.prototxt.txt",
     "confidence_thr": 0.4,
     "min_rel_face_size": 0.01,
     "padding": 0.2
    }
   },
   "landmarks": {
    "ADNet": {
     "model_path": "models/face_landmark_estimation/ADNet.onnx"
    }
   },
   "measures": {
    "UnifiedQualityScore": {
     "model_path": "models/unified_quality_score/magface_iresnet50_norm.onnx"
    },
    "HeadPose": {
     "model_path": "models/head_pose_estimation/mb1_120x120.onnx"
    },
    "FaceOcclusionSegmentation": {
     "model_path": "models/face_occlusion_segmentation/face_occlusion_segmentation_ort.onnx"
    },
    "FaceParsing": {
     "model_path": "models/face_parsing/bisenet_400.onnx"
    }
   }
  }
 }
}
```

OFIQ's C/C++ library provides the class Configuration which is responsible for reading and managing JAXN configurations.

Note that the model paths are specified as paths relative to the directory of the JAXN configuration file. We assume that the file above is stored in <OFIQ-SOURCE>/data.

1.6.1 Configuration of the face detector

The face detector (SSD) must be configured explicitly:

```
{
 "config": {
   "detector": "ssd",
   ...
 }
}
```

Additionally, the path to the model file and other parameters need to be configured:

```
{
 ...
   "params": {
    "detector": {
      "ssd": {
        "model_path": "models/face_detection/ssd_facedetect.caffemodel",
        "prototxt_path": "models/face_detection/ssd_facedetect.prototxt.txt",
        "confidence_thr": 0.4,
        "min_rel_face_size": 0.01,
        "padding": 0.2
      }
    },
  ...
 }
```

A documentation on the parameters are given in the following table.

Parameter	Description
model_path	path to the SSD model file in CAFEE format
prototxt_path	path to SSD's CAFFE protype file
confidence_thr	minimum value for the confidence the detected faces; detected faces with a lower confidence are discarded. Note, the specified value 0.4 (fixed for OFIQ) has been determined experimentally.
min_rel_face_size	the minimum width of the face bounding boxes relative to the width w of the input image; detected faces, with a bounding box width smaller than min_rel_face_size$*$w are discarded. Note, the specified value 0.01 (fixed for OFIQ) has been determined experimentally.
padding	horizontal and vertical padding of the original image prior face detection. Note, the specified value 0.2 (fixed for OFIQ) has been determined experimentally.

1.6.2 Configuration of the landmark extractor

The face landmark extractor (ADNet) must be configured explicitly:

```
{
 "config": {
   ...
   "landmarks": "ADNet",
   ...
 }
}
```

Additionally, the path to the ADNet model file in ONNX format needs to be configured:

```
{
 ...
   "params": {
    ...
    "landmarks": {
      "ADNet": {
        "model_path": "models/face_landmark_estimation/ADNet.onnx"
      }
    },
 ...
}
```

1.6.3 Other required configurations

As suggested by the minimal configuration file given in Configuration, there are model files that need to be configured and their configurations cannot be omitted. These model files are detailed in the following table and shall be configured in the "config"."measures" environment.

Algorithm	Description
HeadPose	Head pose angles are pre-processed and used by some measures; therefore, the measure shall be configured. The path to the 3DDFAV2 model file in ONNX format should be set using the `model_path` key.
FaceOcclusionSegmentation	Face occlusion segmentation pre-processing used by some measures assessing occlusion prevention. The path to the `FaceExtraction` model file in ONNX format should be set using the `model_path` key. NOTE: The OFIQ development team has been permitted by the FaceExtraction authors for inclusion of the model in OFIQ without any restrictions; therefore, the referenced ONNX model file is subject to the OFIQ license agreement.
FaceParsing	Face parsing is pre-processed and used by some measures; therefore, the measure shall be configured (even if no measure is requested that uses the pre-processing result). The path to the `BiSeNet` model file in ONNX format should be set using the `model_path` key.

1.6.4 Requesting measures

OFIQ implements a variety of measures for assessing properties of a facial image. For a measure to be executed by OFIQ, it must be explicitly requested. For example, to only request the unified quality score measure, one can request it as follows.

```
{
 "config": {
```

```
          ...
        "measures": [
         "UnifiedQualityScore"
        ],
          ...
      }
    }
```

At least one measure must be requested. An empty request list will result in OFIQ throwing an error. A full list of requestable measures and its *measure keys* can be found in the table of the default configuration section.

1.6.5 Default configuration

OFIQ is the reference implementation for the ISO/IEC 29794-5 standard. To reproduce the conformance tests of the ISO/IEC 29794-5 standard one should use the (default) configuration provided by the file $<OFIQ-\hookleftarrow$ SOURCE$>$/data/ofiq_config.jaxn; other configurations can be used; however, the resulting outputs of the quality assessment may not comply with the ISO/IEC 29794-5 standard.

The entries of the default configuration are documented in the following table. Details on the algorithms can be found in the ISO/IEC 29794-5 document. Details on the implementations can be found in the OFIQ source code. Note, the QAA identifiers listed in the table are defined in ISO/IEC 29794-5.

QAA identifier	description	config key	request key	configuration parameters	supports quality mapping config? - see details here
-	Face detector	"config">"params".-"detector"	-	see here	-
-	Face landmark estimator	"config">"params".-"landmarks"	-	see here	-
-	Face parsing	"config". "params". "measures". "FaceParsing"	-	see here	-
-	Face occlusion segmentation	"config". "params". "measures". "Face$\hookleftarrow$ Occlusion$\hookleftarrow$ Segmentation"	-	see here	-
-	Landmarked region	"config". "params". "measures". "FaceRegion"	-	alpha: is 0 per default and only used for internal purposes	-
0x41	Unified quality score	"config". "params". "measures". "Unified$\hookleftarrow$ QualityScore"	"config". "measures". "Unified$\hookleftarrow$ QualityScore"	model_$\hookleftarrow$ path: Path to an iResNet50 model file in ONNX format	yes
0x42	Background uniformity	"config". "params". "measures". "Background$\hookleftarrow$ Uniformity"	"config". "measures". "Background$\hookleftarrow$ Uniformity"	none	yes

0x43	Illumination un-formity	"config". "params". "measures". "Illumination↩ Uniformity"	"config". "measures". "Illumination↩ Uniformity"	none	yes
0x44	Luminance brightness	"config". "params". "measures". "Luminance"	"config". "measures". "Luminance"	none	yes
0x45	Luminance contrast	"config". "params". "measures". "Luminance"	"config". "measures". "Luminance"	none	yes
0x46	Abscence of under-exposure	"config". "params". "measures". "Under↩ Exposure↩ Prevention"	"config". "measures". "Under↩ Exposure↩ Prevention"	none	yes
0x47	Abscence of over-exposure	"config". "params". "measures". "Over↩ Exposure↩ Prevention"	"config". "measures". "Over↩ Exposure↩ Prevention"	none	yes
0x48	Pixel intensity variation	"config". "params". "measures". "Dynamic↩ Range"	"config". "measures". "Dynamic↩ Range"	none	yes
0x49	Sharpness	"config". "params". "measures". "Sharpness"	"config". "measures". "Sharpness"	`model_↩ path`: Path to the random forest model file	yes
0x4A	Abscence of compression artifacts	"config". "params". "measures". "No↩ Compression↩ Artifacts"	"config". "measures". "No↩ Compression↩ Artifacts"	`model_↩ path`: Path to OFIQ's compression artifact CNN in ONNX format	yes
0x4B	Colour natural-ity	"config". "params". "measures". "NaturalColour"	"config". "measures". "NaturalColour"	none	yes
0x4C	Face unique-ness	"config". "params". "measures". "SingleFace↩ Present"	"config". "measures". "SingleFace↩ Present"	none	no
0x4D	Eyes openess	"config". "params". "measures". "EyesOpen"	"config". "measures". "EyesOpen"	none	yes
0x4E	Mouth closed-ness	"config". "params". "measures". "MouthClosed"	"config". "measures". "MouthClosed"	none	yes

0x4F	Eyes visibility	"config". "params". "measures". "EyesVisible"	"config". "measures". "EyesVisible"	none	yes
0x50	Mouth occlusion prevention	"config". "params". "measures". "Mouth↩ Occlusion↩ Prevention"	"config". "measures". "Mouth↩ Occlusion↩ Prevention"	none	yes
0x51	Face occlusion prevention	"config". "params". "measures". "Face↩ Occlusion↩ Prevention"	"config". "measures". "Face↩ Occlusion↩ Prevention"	none	yes
0x52	Inter-eye distance length	"config". "params". "measures". "InterEye↩ Distance"	"config". "measures". "InterEye↩ Distance"	none	yes
0x53	Size of the head in the image	"config". "params". "measures". "HeadSize"	"config". "measures". "HeadSize"	none	yes - the argument to the quality mapping is $\|x - 0.45\|$ where x is the native quality score
0x54	Leftward crop of the face image	"config">"params". "measures". "CropOfThe↩ FaceImage". "Leftward↩ CropOfThe↩ FaceImage"	"config". "measures". "CropOfThe↩ FaceImage"	none	yes
0x55	Rightward crop of the face image	"config">"params". "measures". "CropOfThe↩ FaceImage". "Rightward↩ CropOfThe↩ FaceImage"	"config". "measures". "CropOfThe↩ FaceImage"	none	yes
0x56	Upward crop of the face image	"config">"params". "measures". "CropOfThe↩ FaceImage". "UpwardCrop↩ OfTheFace↩ Image"	"config". "measures". "CropOfThe↩ FaceImage"	none	yes
0x57	Downward crop of the face image	"config">"params". "measures". "CropOfThe↩ FaceImage". "Downward↩ CropOfThe↩ FaceImage"	"config". "measures". "CropOfThe↩ FaceImage"	none	yes

0x58	Pose angle yaw frontal alignment	"config">-	"config". "measures". "HeadPose"	none	no
0x59	Pose angle pitch frontal alignment	-	"config". "measures". "HeadPose"	none	no
0x5A	Pose angle roll frontal alignment	-	"config". "measures". "HeadPose"	none	no
0x5B	Expression neutrality	"config">"params"."measures". "Expression↵Neutrality"	"config". "measures". "Expression↵Neutrality"	`cnn_model↵_path1:` Path to the CNN model *enet_↵b0_8_best_↵vgaf_embed↵_zeroed.onnx* derived from here in ONNX format `cnn_model↵_path2:` Path to the CNN model `enet_b2_↵8_embed_↵zeroed.↵onnx` derived from here in ONNX format. `adaboost↵_model_↵path:` Path to the AdaBoost classifier model file `hse_↵1_2_C_↵adaboost.↵yml.gz` from here	yes
0x5C	Abscence of head coverings	"config">"params"."measures". "NoHead↵Covering"	"config". "measures". "NoHead↵Covering"	`threshold` - threshold between 0 and 1-/td>	no

1.6.6 Configuration of the quality mapping

Each measure implemented in OFIQ outputs a pair of values. The first value is called *native quality score*. The second value is called *quality component value* which is an integer between 0 (worst quality) and 100 (best quality). The quality component value and is derived from the *native quality score* using a mapping function. Whether this mapping function can be configured for the measure is stated in the table of section Default configuration.

Note, the OFIQ library implements hard-coded default quality mappings as a fallback.

A configurable quality mapping has the form of

$$Q(h, a, s, x, x_0, w) = h \cdot (a + s \cdot \text{sigmoid}(x, x_0, w))$$

where

$$\text{sigmoid}(x, x_0, w) = (1 + \exp((x_0 - x)/w)^{-1}.$$

Note, x is the native quality score which is mapped to the quality component value. The other symbols denote parameters that can be configured (see the example at the end of this section).

parameter	description	default value
"h"	scale factor	100
"a"	constant shift	0
"s"	signed weight for sigmoid part	1
"x0"	center point within sigmoid function; the default value has been chosen arbitrarily and should specified when a mapping is configured.	4
"w"	divisor within the sigmoid function; the default value has been chosen arbitrarily and should specified when a mapping is configured.	0.7
"round"	applies the compiler's native rounding function (`std::round`) such that only integer values are used as the quality value	true

All parameters are optional and can be omitted. In this case, the default value is chosen. Note, if a mapping results in a value not within 0 and 100, then a clipping is applied choosing the value 0 or 100 being closest to the mapped value.

For example, OFIQ's configuration for the background uniformity measure looks as follows

```
{
  "config" :
  ...
  "params": {
      ...
      "BackgroundUniformity": {
        "Sigmoid" : {
          "h": 190,
          "a": 1,
          "s": -1,
          "x0": 10,
          "w" : 100,
          "round": true
        }
      }
      ...
  }
}
```

1.7 C++ API

To use OFIQ in a C++ application one needs to include the following header file.

```
include <ofiq_lib.h>
```

In the following, we assume that the namespace OFIQ and OFIQ_LIB are used.

```
using namespace OFIQ;
using namespace OFIQ_LIB;
```

An OFIQ instance is initialized using the Interface class as follows.

```
// Get implementation pointer
auto implPtr = Interface::getImplementation();
// Initialization
auto ret = implPtr->initialize(configDir,configFile);
```

Here `configDir` is a `std::string`-representation of the path to the directory in which a JAXN configuration file of name `configFile` is stored - as documented in the configuration section. Note, that the path can be absolute or relative to the path of the current working directory.

The input image is read by using the readImage function as follows

```
Image image;
ReturnStatus retStatus = readImage(imagePath, image);
```

where `imagePath` is a `std::string`-representation of a path to an image file. The representation is written to the `image` object of type Image.

To compute the quality assessments, run

```
FaceImageQualityAssessment assessment;
ReturnStatus retStatus = implPtr->vectorQuality(image, assessment);
```

A successful computation is indicated by `retStatus.code` if it is of value ReturnCode::Success. Then the assessment result is stored in a FaceImageQualityAssessment struct object. The obtained FaceImageQualityAssessment object has a `std::map` member which, for a specified QualityMeasure key, returns the QualityMeasureResult. A QualityMeasureResult struct object contains the native quality score stored in the rawScore member and the quality component value stored in the scalar member. Note, both members are encoded as a `double` values although the scalar member should (on successful quality measure computation) be an integer value between 0 and 100. To check whether a QualityMeasureResult has been computed successfully, one checks if its code member agrees with the value QualityMeasureReturnCode::Success.

1.8 Implementation and pre-processing workflow

Quality assessment is controlled by the implementation of the OFIQImpl class. A shared pointer to an OFIQImpl object is returned by the Interface::getImplementation() function. The implementation needs to be initialized once using the OFIQImpl::initialize() function. Note, the OFIQImpl::initialize() function loads all model files as specified in the input configuration into memory; thus, one should avoid creating repeated instances of the OFIQImpl.

After successful initialization, the implementation object can be used and one can repeatedly invoke the OFIQImpl::vectorQuality() function to assess the quality of a series of facial images.

The internal workflow of the OFIQImpl::vectorQuality() implementation is as follows.

1. Pre-processing of the input image using the OFIQImpl::performPreprocessing() function.

   (a) Face detection implemented by SSDFaceDetector::UpdateFaces().

   (b) Pose estimation implemented by HeadPose3DDFAV2::updatePose().

   (c) Landmark extraction implemented by ADNetFaceLandmarkExtractor::updateLandmarks().

   (d) Facial alignment implemented by OFIQImpl::alignFaceImage().

   (e) Face parsing implemented by FaceParsing::UpdateMask().

   (f) Face occlusion segmentation implemented by FaceOcclusionSegmentation::UpdateMask().

2. Quality assessment using the Executor::ExecuteAll() function: For all requested measures

   (a) its Execute()

   (b) and then its SetQualityMeasure() functions are invoked.

1.9 Release notes

This is OFIQ Version 1.0.0-RC (2024-03-15). The following table lists all measures and its implementation provided by this release of OFIQ. Details on the configuration and on requesting measures can be found here. Note, the QAA identifiers listed in the table are defined in ISO/IEC 29794-5.

QAA identifier	Description	OFIQ implementation reference
0x41	`MagFace`-based unified quality score measure.	UnifiedQualityScore
0x42	Gradient-based background uniformity.	BackgroundUniformity
0x43	Illumination unformity by summing up the minima of the histograms of the left and the right side of the face.	IlluminationUniformity
0x44	Luminance mean measure computed from the luminance histogram	Luminance
0x45	Luminance variance measure computed from the luminance histogram	Luminance
0x46	Under-exposure prevention by computing the proportion of low-intensity pixels in the luminance image to assess the abscence of under-exposure	UnderExposurePrevention
0x47	Over-exposure prevention by computing the proportion of high-intensity pixels in the luminance image to assess the abscence of over-exposur	OverExposurePrevention
0x48	Dynamic range computed from the luminance histogram.	DynamicRange
0x49	Sharpness assessment based on a random forest classifier trained by the OFIQ development team.	Sharpness
0x4A	Assessment of the absence of compression artifact (both JPEG and JPEG2000) based on a CNN trained by the OFIQ development team.	CompressionArtifacts
0x4B	Assessment of the naturalness of the colour based on the conversion of the RGB presentation of the image to the CIELAB colour space.	NaturalColour
0x4C	Assessment of the uniqueness of the most dominant face detected by comparing its size with the size of the second largest face detected	SingleFacePresent
0x4D	Eyes openness assessment based on computing eyes aspect ratio from eye landmarks	EyesOpen

0x4E	Mouth closed assessment based on computing a ratio from mouth landmarks	MouthClosed
0x4F	Eyes visibility assessment by measuring the coverage of the eye visibility zone with the result of face occlusion segmentation computed during pre-processing.	EyesVisible
0x50	Assessment of the absence of mouth occlusion by measuring the coverage of the mouth region from mouth landmarks with the result of face occlusion segmentation computed on pre-processing.	MouthOcclusionPrevention
0x51	Assessment of the absence of face occlusion by measuring the coverage of the landmarked region with the result of face occlusion segmentation computed during pre-processing.	FaceOcclusionPrevention
0x52	Inter-eye distance assessment based on computing the Euclidean length of eyes' centres and multiplication with the secant of the yaw angle computed during pre-processing.	InterEyeDistance
0x53	Size of the head based on computing the height of the face computed from facial landmarks with the height of the image.	HeadSize
0x54	Leftward crop of the face image	CropOfTheFaceImage
0x55	Rightward crop of the face image	CropOfTheFaceImage
0x56	Downward crop of the face image	CropOfTheFaceImage
0x57	Upward crop of the face image	CropOfTheFaceImage
0x58	Pose angle yaw frontal alignment based on the `3DDFAV2`.	HeadPose
0x59	Pose angle pitch frontal alignment based on the `3DDFAV2`	HeadPose
0x5A	Pose angle roll frontal alignment based on the `3DDFAV2`	HeadPose
0x5B	Expression neutrality estimation based on a fusion of `HSEMotion` with with `Efficient-Expression-Neutrality-Estimation`.	ExpressionNeutrality
0x5C	Assessment of the absence of head coverings by counting the pixels being labeled as head covers in the mask output by the face parsing computed during pre-processing.	NoHeadCoverings

1.9.1 Changelog

1.9.1.1 Version 1.0.0-RC (2024-03-15)

Initial release of OFIQ's release candidate providing the measures described in the Release notes

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

Chapter 4

Class Index

4.1 Class List

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

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

Chapter 6

Namespace Documentation

6.1 cv Namespace Reference

OpenCV's namespace.

6.1.1 Detailed Description

OpenCV's namespace.

6.2 OFIQ Namespace Reference

Namespace for OFIQ API.

Classes

- struct BoundingBox

  Data structure for descibing bounding boxes, e.g. the face region of the faces found by a face detector.
- struct FaceImageQualityAssessment

  Data structure storing the results of the different measurement computations.
- struct FaceLandmarks

  Data structure for storing facial landmarks.
- struct Image

  Struct representing a single image.
- class Interface

  The interface to FACE QA implementation.
- struct LandmarkPoint

  Data structure to describe the x and y coordinate of a landmark.
- struct QualityMeasureResult

  Data structure to handle the results of a quality measure.
- struct ReturnStatus

  A structure to contain information about a failure by the software under test.

Typedefs

- using QualityAssessments = std::map<QualityMeasure, QualityMeasureResult>

    Data structure that stores key-value pairs, with each entry representing a quality element and its value.
- using Landmarks = std::vector<LandmarkPoint>

    container for a collection of landmarks, e.g. belonging to all the landmarks detected on a face image.

Enumerations

- enum class ReturnCode {
    Success = 0 , ImageReadingError , ImageWritingError , MissingConfigParamError ,
    UnknownConfigParamError , FaceDetectionError , FaceLandmarkExtractionError , FaceOcclusionSegmentationError
    ,
    FaceParsingError , UnknownError , QualityAssessmentError , NotImplemented }

    Return codes for functions specified in this API.
- enum class QualityMeasure {
    UnifiedQualityScore = 0x41 , BackgroundUniformity = 0x42 , IlluminationUniformity = 0x43 , Luminance =
    -0x44 ,
    LuminanceMean = 0x44 , LuminanceVariance = 0x45 , UnderExposurePrevention = 0x46 , OverExposurePrevention
    = 0x47 ,
    DynamicRange = 0x48 , Sharpness = 0x49 , CompressionArtifacts = 0x4a , NaturalColour = 0x4b ,
    SingleFacePresent = 0x4c , EyesOpen = 0x4d , MouthClosed = 0x4e , EyesVisible = 0x4f ,
    MouthOcclusionPrevention = 0x50 , FaceOcclusionPrevention = 0x51 , InterEyeDistance = 0x52 , HeadSize
    = 0x53 ,
    CropOfTheFaceImage = -0x54 , LeftwardCropOfTheFaceImage = 0x54 , RightwardCropOfTheFaceImage =
    0x55 , DownwardCropOfTheFaceImage = 0x56 ,
    UpwardCropOfTheFaceImage = 0x57 , HeadPose = -0x58 , HeadPoseYaw = 0x58 , HeadPosePitch = 0x59 ,
    HeadPoseRoll = 0x5a , ExpressionNeutrality = 0x5b , NoHeadCoverings = 0x5c , NotSet = -1 }

    Enums presenting the measure labels.
- enum class QualityMeasureReturnCode { Success = 0 , FailureToAssess , NotInitialized }

    Return codes for QualityMeasureResult.
- enum class FaceDetectorType { OPENCVSSD , NotSet }

    Enum describing the different face detector implementations.
- enum class LandmarkType { LM_98 , NotSet }

    Enum describing the different implementations of landmarks.

Functions

- std::ostream & operator<< (std::ostream &s, const ReturnCode &rc)

6.2.1 Detailed Description

Namespace for OFIQ API.

Namespace for OFIQ API.

6.2.2 Typedef Documentation

6.2.2.1 Landmarks

```
using OFIQ::Landmarks = std::vector<LandmarkPoint>
```

container for a collection of landmarks, e.g. belonging to all the landmarks detected on a face image.

6.2.2.2 QualityAssessments

using OFIQ::QualityAssessments = std::map<QualityMeasure, QualityMeasureResult>

Data structure that stores key-value pairs, with each entry representing a quality element and its value.

6.2.3 Enumeration Type Documentation

6.2.3.1 FaceDetectorType

enum class OFIQ::FaceDetectorType [strong]

Enum describing the different face detector implementations.

Enumerator

OPENCVSSD	face detector based on the ssd implementation in opencv.
NotSet	unknown face detector

6.2.3.2 LandmarkType

enum class OFIQ::LandmarkType [strong]

Enum describing the different implementations of landmarks.

Enumerator

LM_98	Landmarks extracted with the adnet detector.
NotSet	used for iterating through the enums.

6.2.3.3 QualityMeasure

enum class OFIQ::QualityMeasure [strong]

Enums presenting the measure labels.

Enumerator

UnifiedQualityScore	UnifiedQualityScore
BackgroundUniformity	BackgroundUniformity
IlluminationUniformity	IlluminationUniformity
Luminance	the common measure implementation for LuminanceMean, LuminanceVariance
LuminanceMean	LuminanceMean
LuminanceVariance	LuminanceVariance
UnderExposurePrevention	UnderExposurePrevention
OverExposurePrevention	OverExposurePrevention

Enumerator

DynamicRange	DynamicRange
Sharpness	Sharpness
CompressionArtifacts	CompressionArtifacts
NaturalColour	NaturalColour
SingleFacePresent	SingleFacePresent
EyesOpen	EyesOpen
MouthClosed	MouthClosed
EyesVisible	EyesVisible
MouthOcclusionPrevention	MouthOcclusionPrevention
FaceOcclusionPrevention	FaceOcclusionPrevention
InterEyeDistance	InterEyeDistance
HeadSize	HeadSize
CropOfTheFaceImage	CropOfTheFaceImage: common measure for {Left,Right,Up,Down}wardCropOfTheFaceImage
LeftwardCropOfTheFaceImage	LeftwardCropOfTheFaceImage
RightwardCropOfTheFaceImage	RightwardCropOfTheFaceImage
DownwardCropOfTheFaceImage	DownwardCropOfTheFaceImage
UpwardCropOfTheFaceImage	UpwardCropOfTheFaceImage
HeadPose	HeadPose
HeadPoseYaw	HeadPoseYaw
HeadPosePitch	HeadPosePitch
HeadPoseRoll	HeadPoseRoll
ExpressionNeutrality	ExpressionNeutrality
NoHeadCoverings	NoHeadCoverings
NotSet	unknown measure

6.2.3.4 QualityMeasureReturnCode

enum class OFIQ::QualityMeasureReturnCode [strong]

Return codes for QualityMeasureResult.

Enumerator

Success	Success
FailureToAssess	Unable to assess a quality measure
NotInitialized	Quality measure is not initialized

6.2.3.5 ReturnCode

enum class OFIQ::ReturnCode [strong]

Return codes for functions specified in this API.

Enumerator

Success	Success
ImageReadingError	Failed to read an image.
ImageWritingError	failed to write an image to disk.
MissingConfigParamError	A required config parameter is missing
UnknownConfigParamError	A required config parameter is missing
FaceDetectionError	Unable to detect a face in the image
FaceLandmarkExtractionError	Unable to extract landmarks from face
FaceOcclusionSegmentationError	Unable to extract occlusion segments from face
FaceParsingError	Unable to parse face
UnknownError	Catch-all error
QualityAssessmentError	Failure to generate a quality score on the input image
NotImplemented	Function is not implemented

6.2.4 Function Documentation

6.2.4.1 operator$<<$()

```
std::ostream & OFIQ::operator<< (
            std::ostream & s,
            const ReturnCode & rc )  [inline]
```

Output stream operator for a ReturnCode object.

6.3 OFIQ_LIB Namespace Reference

Namespace for OFIQ implementations.

Namespaces

- namespace modules

Classes

- class Configuration

  Configuration class.
- class FaceDetectorInterface

  Provides the interface class to the face detector implementations.
- class FaceLandmarkExtractorInterface

  Implements the base class for the face landmark extractors.
- struct NeuronalNetworkContainer

  Neural network container for OFIQ's preprocessing steps.
- class OFIQError

  Implementation of a custom exception.
- class OFIQImpl

Implementation of the OFIQ_LIB.

- struct Point2i

  Representation of a point with integer arithmetics.

- class PoseEstimatorInterface

  Implementation of the base class for integrating pose estimation algorithms capable of estimating three head orientation angles (yaw, pitch and roll) from a face image.

- class SegmentationExtractorInterface

  Base class for the different implementation of segmentation algorithms.

- class Session

Typedefs

- using ExposureRange = std::array<int, 2>
- using EulerAngle = std::array<double, 3>

Functions

- OFIQ_EXPORT OFIQ::ReturnStatus readImage (const std::string &filename, OFIQ::Image &image)

  Read image from disk.

- OFIQ_EXPORT double ColorConvert (double v)

  Converts a color as specified in ISO/IEC 29794-5.

- OFIQ_EXPORT double Cubic (double x, double k, double eps)

  Cubic flattening function.

- OFIQ_EXPORT void ConvertBGRToCIELAB (const cv::Mat &bgrImage, double &a, double &b)

  Computes CIELAB values a^* and b^* from a BGR image.

- OFIQ_EXPORT cv::Mat GetLuminanceImageFromBGR (const cv::Mat &bgrImage)

  Converts a BGR image to the luminance image.

- OFIQ_EXPORT void CalculateReferencePoints (const OFIQ::FaceLandmarks &landmarks, OFIQ::LandmarkPoint &leftEyeCenter, OFIQ::LandmarkPoint &rightEyeCenter, double &interEyeDistance, double &eyeMouth↩Distance)

  Computes the left eye center, the right eye center, the (planar) inter-eye-distance and the eye to mouth distance from facial landmarks.

- OFIQ_EXPORT void CalculateRegionOfInterest (cv::Rect &leftRegionOfInterest, cv::Rect &rightRegionOf↩Interest, const OFIQ::LandmarkPoint &leftEyeCenter, const OFIQ::LandmarkPoint &rightEyeCenter, const double interEyeDistance, const double eyeMouthDistance)

  Extracts regions being of interest for some measures (e.g. NaturalColour).

- OFIQ_EXPORT void GetNormalizedHistogram (const cv::Mat &luminanceImage, const cv::Mat &maskImage, cv::Mat1f &histogram)

  Computes the normalized histogram from a luminance image in 256 chunks.

- OFIQ_EXPORT double CalculateExposure (const Session &session, const ExposureRange &exposure↩Range)

  Helper function for some measures.

- OFIQ_EXPORT double ComputeBrightnessAspect (const cv::Mat &luminanceImage, const cv::Mat &mask↩Image, const ExposureRange &exposureRange)

  Helper function for some measures.

- OFIQ_EXPORT void makeSquareBoundingBoxWithPadding (const OFIQ::BoundingBox &i_bb, const cv::Mat &i_input_image, cv::Mat &o_output_image, OFIQ::BoundingBox &o_bb, Point2i &o_translation_vector)

  Some computations, especially neural networks, need a squarred image as input. This funtion consumes a boundig box and an input image. The greater parameter of width or height is used to define the side length of the new squarred bounding box. The face will be centered in the bounding box. Padding is added if needed. The squarred bounding box is used generate a new cropped image, the o_output_image. Required translations are described by the translation vector o_translation_vector.

- OFIQ_EXPORT OFIQ::BoundingBox makeSquareBoundingBox (const OFIQ::BoundingBox &i_bb)

  This function converts a non-squarred bounding box into an squarred one. The side length is defined by the greater one of height or width.

- OFIQ_EXPORT size_t findLargestBoundingBox (const std::vector< OFIQ::BoundingBox > &faceRects)

  This function returns the position of the largest bounding box (largest in terms of area) from a vector of bounding boxes.

- OFIQ_EXPORT cv::Mat copyToCvImage (const OFIQ::Image &sourceImage, bool asGrayImage=false)

  Convert images in OFIQ::Image format into the OpenCV cv::Mat format. The image can be converted from color to gray scale by setting the parameter asGrayImage to true.

- OFIQ_EXPORT cv::Mat alignImage (const OFIQ::Image &faceImage, const OFIQ::FaceLandmarks &face↩Landmarks, OFIQ::FaceLandmarks &alignedFaceLandmarks, cv::Mat &transformationMatrix)

  This function transforms a face image so that the position of the eyes, nose and mouth are roughly at a pre-defined position. Face alignment is the translation, rotation and scaling of the image to do this.

- OFIQ_EXPORT void calculateEyeCenter (const OFIQ::FaceLandmarks &faceLandmarks, Point2f &leftEye↩Center, Point2f &rightEyeCenter)

  Based on face landmarks the center of the left and right eye are computed.

- OFIQ_EXPORT OFIQ::Image MakeGreyImage (uint16_t width, uint16_t height)

  This function generates a gray scaled image with the resolution passed by the call.

- OFIQ_EXPORT float tmetric (const OFIQ::FaceLandmarks &faceLandmarks)

  Based on the provided landmarks this function computes the distance between the point between the eyes and the chin.

- OFIQ_EXPORT void rotationMatrixToEulerAngles (const cv::Mat &R, std::vector< double > &angles)

  Based on a given rotation matrix this functions computes and returns the corresponding Euler angles.

6.3.1 Detailed Description

Namespace for OFIQ implementations.

Provides implementations in OFIQ.

Namespace for OFIQ implementations.

6.3.2 Typedef Documentation

6.3.2.1 EulerAngle

```
using OFIQ_LIB::EulerAngle = std::array<double, 3>
```

6.3.2.2 ExposureRange

```
using OFIQ_LIB::ExposureRange = std::array<int, 2>
```

6.3.3 Function Documentation

6.3.3.1 alignImage()

```
OFIQ_EXPORT cv::Mat OFIQ_LIB::alignImage (
            const OFIQ::Image & faceImage,
            const OFIQ::FaceLandmarks & faceLandmarks,
            OFIQ::FaceLandmarks & alignedFaceLandmarks,
            cv::Mat & transformationMatrix )
```

This function transforms a face image so that the position of the eyes, nose and mouth are roughly at a pre-defined position. Face alignment is the translation, rotation and scaling of the image to do this.

Parameters

faceImage	Input image.
faceLandmarks	Face landmarks, based on the face represented in the input image.
alignedFaceLandmarks	Face landmarks of the aligned face image.
transformationMatrix	Transformation matrix used to transform the landmarks.

Returns

cv::Mat Aligned face image with a resolution of 616x616.

6.3.3.2 CalculateExposure()

OFIQ_EXPORT double OFIQ_LIB::CalculateExposure (
        const Session & *session,*
        const ExposureRange & *exposureRange*)

Helper function for some measures.

The function is used by UnderExposurePrevention and OverExposurePrevention class. Details can be found in the ISO/IEC 29794-5 standard.

Parameters

session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method
exposureRange	Range of pixels for which the aspect is computed.

Returns

Exposure computed from the inputs.

6.3.3.3 calculateEyeCenter()

OFIQ_EXPORT void OFIQ_LIB::calculateEyeCenter (
        const OFIQ::FaceLandmarks & *faceLandmarks,*
        Point2f & *leftEyeCenter,*
        Point2f & *rightEyeCenter*)

Based on face landmarks the center of the left and right eye are computed.

Parameters

faceLandmarks	Input face landmarks.
leftEyeCenter	Point coordinates of the left eye center.
rightEyeCenter	Point coordinates of the right eye center.

6.3.3.4 CalculateReferencePoints()

```
OFIQ_EXPORT void OFIQ_LIB::CalculateReferencePoints (
            const OFIQ::FaceLandmarks & landmarks,
            OFIQ::LandmarkPoint & leftEyeCenter,
            OFIQ::LandmarkPoint & rightEyeCenter,
            double & interEyeDistance,
            double & eyeMouthDistance )
```

Computes the left eye center, the right eye center, the (planar) inter-eye-distance and the eye to mouth distance from facial landmarks.

Parameters

in	*landmarks*	Facial landmarks
out	*leftEyeCenter*	Left eye center computed from landmarks
out	*rightEyeCenter*	Right eye center computed from landmarks
out	*interEyeDistance*	Inter-eye distance computed from landmarks (does not consider the yaw angle).
out	*eyeMouthDistance*	Distance from the eyes' midpoint to the mouth.

6.3.3.5 CalculateRegionOfInterest()

```
OFIQ_EXPORT void OFIQ_LIB::CalculateRegionOfInterest (
            cv::Rect & leftRegionOfInterest,
            cv::Rect & rightRegionOfInterest,
            const OFIQ::LandmarkPoint & leftEyeCenter,
            const OFIQ::LandmarkPoint & rightEyeCenter,
            const double interEyeDistance,
            const double eyeMouthDistance )
```

Extracts regions being of interest for some measures (e.g. NaturalColour).

Details can be found in the ISO/IEC 29794-5 standard for the Natural colour measure.

Parameters

out	*leftRegionOfInterest*	Rectangular region corresponding to the left eye
out	*rightRegionOfInterest*	Rectangular region corresponding to the right eye
in	*leftEyeCenter*	Center of the left eye
in	*rightEyeCenter*	Center of the right eye
in	*interEyeDistance*	Planar inter-eye distance
in	*eyeMouthDistance*	Distance from the eyes' centers midpoint to the mouth

Returns

Applies a heuristic to estimate two regions being of interest for the natural colour measure.

6.3.3.6 ColorConvert()

```
OFIQ_EXPORT double OFIQ_LIB::ColorConvert (
            double v )
```

Converts a color as specified in ISO/IEC 29794-5.

Parameters

v	An intensity value between 0 (black) and 1 (white).

Returns

If $v > 0.04045$, then $((v + 0.055)/1.055)^{2.4}$ is returned; otherwise, if $v \leq 0.04045$, then $v/12.92$ is returned.

6.3.3.7 ComputeBrightnessAspect()

```
OFIQ_EXPORT double OFIQ_LIB::ComputeBrightnessAspect (
            const cv::Mat & luminanceImage,
            const cv::Mat & maskImage,
            const ExposureRange & exposureRange )
```

Helper function for some measures.

The function is used by UnderExposurePrevention and OverExposurePrevention class. Details can be found in the ISO/IEC 29794-5 standard.

Parameters

luminanceImage	luminance image.
maskImage	The mask on which the aspect is computed
exposureRange	Range of pixels for which the aspect is computed.

Returns

Brightness aspect computed from the inputs.

6.3.3.8 ConvertBGRToCIELAB()

```
OFIQ_EXPORT void OFIQ_LIB::ConvertBGRToCIELAB (
            const cv::Mat & bgrImage,
            double & a,
            double & b )
```

Computes CIELAB values a^* and b^* from a BGR image.

Parameters

in	bgrImage	BGR image
out	a	CIELAB value a^*
out	b	CIELAB value b^*

6.3.3.9 copyToCvImage()

```
OFIQ_EXPORT cv::Mat OFIQ_LIB::copyToCvImage (
            const OFIQ::Image & sourceImage,
            bool asGrayImage = false )
```

Convert images in OFIQ::Image format into the OpenCV cv::Mat format. The image can be converted from color to gray scale by setting the parameter asGrayImage to true.

Parameters

sourceImage	Input image.
asGrayImage	Switch for adding gray scale conversion.

Returns

cv::Mat Input image in cv::Mat format.

6.3.3.10 Cubic()

```
OFIQ_EXPORT double OFIQ_LIB::Cubic (
            double x,
            double k,
            double eps )
```

Cubic flattening function.

Parameters

x	Argument
k	Argument
eps	Argument ϵ

Returns

If $x \leq \epsilon$, then $(k \cdot x + 16)/116$ is returned; otherwise, if $x > \epsilon$, then $\sqrt[3]{x}$ is returned.

6.3.3.11 findLargestBoundingBox()

```
OFIQ_EXPORT size_t OFIQ_LIB::findLargestBoundingBox (
            const std::vector< OFIQ::BoundingBox > & faceRects )
```

This function returns the position of the largest bounding box (largest in terms of area) from a vector of bounding boxes.

Parameters

faceRects	Vector containing bounding boxes.

Returns

> size_t Position of the largest bounding box in the vector.

6.3.3.12 GetLuminanceImageFromBGR()

```
OFIQ_EXPORT cv::Mat OFIQ_LIB::GetLuminanceImageFromBGR (
            const cv::Mat & bgrImage )
```

Converts a BGR image to the luminance image.

The conversion is specified in the ISO/IEC 29794-5 standard and uses the function ColorConvert() .

Parameters

bgrImage	BGR image

Returns

> Luminance image.

6.3.3.13 GetNormalizedHistogram()

```
OFIQ_EXPORT void OFIQ_LIB::GetNormalizedHistogram (
            const cv::Mat & luminanceImage,
            const cv::Mat & maskImage,
            cv::Mat1f & histogram )
```

Computes the normalized histogram from a luminance image in 256 chunks.

Parameters

in	*luminanceImage*	Luminance image as returned by GetLuminanceImageFromBGR() .
in	*maskImage*	The histogram is computed on pixels where the values of maskImage are non-zero.
out	*histogram*	Array of length 256 where the histogram is stored.

6.3.3.14 MakeGreyImage()

```
OFIQ_EXPORT OFIQ::Image OFIQ_LIB::MakeGreyImage (
            uint16_t width,
            uint16_t height )
```

This function generates a gray scaled image with the resolution passed by the call.

Parameters

width	Width of the generated image.
height	Height of the generated image.

Returns

OFIQ::Image Generated gray scaled image.

6.3.3.15 makeSquareBoundingBox()

OFIQ_EXPORT OFIQ::BoundingBox OFIQ_LIB::makeSquareBoundingBox (
            const OFIQ::BoundingBox & *i_bb*)

This function converts a non-squarred bounding box into an squarred one. The side length is defined by the greater one of height or width.

Parameters

i_bb	Input bounding box.

Returns

OFIQ::BoundingBox Squarred bounding box.

6.3.3.16 makeSquareBoundingBoxWithPadding()

OFIQ_EXPORT void OFIQ_LIB::makeSquareBoundingBoxWithPadding (
            const OFIQ::BoundingBox & *i_bb,*
            const cv::Mat & *i_input_image,*
            cv::Mat & *o_output_image,*
            OFIQ::BoundingBox & *o_bb,*
            Point2i & *o_translation_vector*)

Some computations, especially neural networks, need a squarred image as input. This funtion consumes a boundig box and an input image. The greater parameter of width or height is used to define the side length of the new squarred bounding box. The face will be centered in the bounding box. Padding is added if needed. The squarred bounding box is used generate a new cropped image, the o_output_image. Required translations are described by the translation vector o_translation_vector.

Parameters

i_bb	Initial bounding box.
i_input_image	Input image.
o_output_image	Cropped output image. Cropping is based on the computed squarred bounding box.
o_bb	Squarred bounding box.
o_translation_vector	Translation vector.

6.3.3.17 readImage()

OFIQ_EXPORT OFIQ::ReturnStatus OFIQ_LIB::readImage (
            const std::string & *filename,*
            OFIQ::Image & *image*)

Read image from disk.

Parameters

in	*filename*	Path and file name of the image being read from disk.
out	*image*	Reference to the image object where the data is loaded to.

Returns

OFIQ::ReturnStatus

6.3.3.18 rotationMatrixToEulerAngles()

```
OFIQ_EXPORT void OFIQ_LIB::rotationMatrixToEulerAngles (
            const cv::Mat & R,
            std::vector< double > & angles )
```

Based on a given rotation matrix this functions computes and returns the corresponding Euler angles.

Parameters

R	Input rotation matrix
angles	Container with the computed Euler angles for [x,y,z] axis

6.3.3.19 tmetric()

```
OFIQ_EXPORT float OFIQ_LIB::tmetric (
            const OFIQ::FaceLandmarks & faceLandmarks )
```

Based on the provided landmarks this function computes the distance between the point between the eyes and the chin.

Parameters

faceLandmarks	Input face landmarks.

Returns

float Computed distance.

6.4 OFIQ_LIB::modules Namespace Reference

Namespaces

- namespace detectors

    Provides face detector implementations.

- namespace landmarks

    Provides implementations of a landmark extractors.

- namespace measures

    Provides measures implemented in OFIQ.
- namespace poseEstimators

    Provides implementation of a head pose estimator.
- namespace segmentations

    Provides segmentation-related implementations.

6.5 OFIQ_LIB::modules::detectors Namespace Reference

Provides face detector implementations.

Classes

- class SSDFaceDetector

    Implementation of a face detector using the SSD face detector CNN.

6.5.1 Detailed Description

Provides face detector implementations.

6.6 OFIQ_LIB::modules::landmarks Namespace Reference

Provides implementations of a landmark extractors.

Namespaces

- namespace adnet

    Namespace for ADNet-specific landmarks.

Classes

- class ADNetFaceLandmarkExtractor

    Class implementing the FaceLandmarkExtractorInterface interface.
- class FaceMeasures

    Provides static functions doing computations with landmarks.
- struct LandmarkPair

    Data container for storing pairs of landmarks.
- class PartExtractor

    Class that provides helper methods for the administration of landmarks.

Typedefs

- using LandmarkId = int

    Type definition of a landmark index.

- using LandmarkIds = std::vector<LandmarkId>

    Type definition of a list of landmark indices.

- using FaceMap = std::map<FaceParts, LandmarkIds>

    Type definition of a face map to access landmark indices for a queried face part.

- using LandmarkIdPair = std::array<LandmarkId, 2>

    Type definition for a pair of landmark index.

- using LandmarkIdPairs = std::vector<LandmarkIdPair>

    Type definition for a list of landmark index pairs.

- using FacePairMap = std::map<FaceParts, LandmarkIdPairs>

    Structure defining pairs of landmark indices.

Enumerations

- enum class FaceParts {
  LEFT_EYE , RIGHT_EYE , LEFT_EYE_CORNERS , RIGHT_EYE_CORNERS ,
  MOUTH_OUTER , MOUTH_INNER , FACE_CONTOUR , MOUTH_CENTER ,
  CHIN , NOSETIP , FOREHEAD }

    Enumeration of facial landmark parts.

6.6.1   Detailed Description

Provides implementations of a landmark extractors.

Provides implementations for computations with landmarks.

Provides implementations related to facial landmarks.

6.6.2   Typedef Documentation

6.6.2.1   FaceMap

using OFIQ_LIB::modules::landmarks::FaceMap = std::map<FaceParts, LandmarkIds>

Type definition of a face map to access landmark indices for a queried face part.

6.6.2.2   FacePairMap

using OFIQ_LIB::modules::landmarks::FacePairMap = std::map<FaceParts, LandmarkIdPairs>

Structure defining pairs of landmark indices.

6.6.2.3 LandmarkId

using OFIQ_LIB::modules::landmarks::LandmarkId = int

Type definition of a landmark index.

6.6.2.4 LandmarkIdPair

using OFIQ_LIB::modules::landmarks::LandmarkIdPair = std::array<LandmarkId, 2>

Type definition for a pair of landmark index.

6.6.2.5 LandmarkIdPairs

using OFIQ_LIB::modules::landmarks::LandmarkIdPairs = std::vector<LandmarkIdPair>

Type definition for a list of landmark index pairs.

6.6.2.6 LandmarkIds

using OFIQ_LIB::modules::landmarks::LandmarkIds = std::vector<LandmarkId>

Type definition of a list of landmark indices.

6.6.3 Enumeration Type Documentation

6.6.3.1 FaceParts

enum class OFIQ_LIB::modules::landmarks::FaceParts [strong]

Enumeration of facial landmark parts.

Enumerator

LEFT_EYE	left as seen on the image, it's the persons right eye
RIGHT_EYE	right as seen on the image, it's the persons left eye
LEFT_EYE_CORNERS	two landmarks - outer, inner corner
RIGHT_EYE_CORNERS	two landmarks - outer, inner corner
MOUTH_OUTER	outer landmarks of mouth
MOUTH_INNER	inner landmarks of mouth
FACE_CONTOUR	contour of the face
MOUTH_CENTER	center of the mouth
CHIN	chin
NOSETIP	nosetip
FOREHEAD	forehead

6.7   OFIQ_LIB::modules::landmarks::adnet Namespace Reference

Namespace for ADNet-specific landmarks.

Variables

- const LandmarkIds leftEye {60,61,62,63,64,65,66,67}

  Landmark indices (ADNet) of the left eye.
- const LandmarkIds rightEye {68,69,70,71,72,73,74,75}

  Landmark indices (ADNet) of the right eye.
- const LandmarkIds leftEyeCorners {60,64}

  Landmark indices (ADNet) of the left eyes' corners.
- const LandmarkIds rightEyeCorners {68,72}

  Landmark indices (ADNet) of the right eyes' corners.
- const LandmarkIds nosetip {54}

  Landmark index (ADNet) of the nose tip.
- const LandmarkIds mouthOuter {76,77,78,79,80,81,82,83,84,85,86,87}

  Landmark indices (ADNet) on the mouth's outer contour.
- const LandmarkIds mouthInner {88,89,90,91,92,93,94,95}

  Landmark indices (ADNet) on the mouth's inner lip borders.
- const LandmarkIds contour {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32}

  Landmark indices (ADNet) of the face contour.
- const LandmarkIds forehead {}

  Landmark indices (ADNet) of the forehead (empty for ADNet).
- const LandmarkIds chin {16}

  Landmark index (ADNet) of the chin.
- const landmarks::FaceMap FaceMap

  ADNets face map definition.
- const LandmarkIdPairs pairsLeftEye

  Pair indices of landmarks (ADNet) for the left eye.
- const LandmarkIdPairs pairsRightEye

  Landmark index pairs (ADNet) of landmarks for the right eye.
- const LandmarkIdPairs pairsInnerLip

  Landmark index pairs (ADNet) of inner lip pairs.
- const LandmarkIdPairs pairsMouthCenter

  Landmark index pair (ADNet) of the inner mouth (lips) center.
- const landmarks::FacePairMap FacePairMap

  ADNets face pair map definition.

6.7.1   Detailed Description

Namespace for ADNet-specific landmarks.

6.7.2   Variable Documentation

6.7.2.1   chin

```
const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::chin {16}
```

Landmark index (ADNet) of the chin.

6.7.2.2 contour

const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::contour {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18

Landmark indices (ADNet) of the face contour.

6.7.2.3 FaceMap

const landmarks::FaceMap OFIQ_LIB::modules::landmarks::adnet::FaceMap

Initial value:
```
{
        {FaceParts::LEFT_EYE,          leftEye        },
        {FaceParts::RIGHT_EYE,         rightEye       },
        {FaceParts::LEFT_EYE_CORNERS,  leftEyeCorners},
        {FaceParts::RIGHT_EYE_CORNERS, rightEyeCorners},
        {FaceParts::MOUTH_OUTER,       mouthOuter     },
        {FaceParts::MOUTH_INNER,       mouthInner     },
        {FaceParts::FACE_CONTOUR,      contour        },
        {FaceParts::CHIN,              chin           },
        {FaceParts::NOSETIP,           nosetip        },
        {FaceParts::FOREHEAD,          forehead       }
    }
```

ADNets face map definition.

6.7.2.4 FacePairMap

const landmarks::FacePairMap OFIQ_LIB::modules::landmarks::adnet::FacePairMap

Initial value:
```
{
        {FaceParts::LEFT_EYE,     pairsLeftEye     },
        {FaceParts::RIGHT_EYE,    pairsRightEye    },
        {FaceParts::MOUTH_INNER,  pairsInnerLip    },
        {FaceParts::MOUTH_CENTER, pairsMouthCenter}
    }
```

ADNets face pair map definition.

6.7.2.5 forehead

const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::forehead {}

Landmark indices (ADNet) of the forehead (empty for ADNet).

6.7.2.6 leftEye

const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::leftEye {60,61,62,63,64,65,66,67}

Landmark indices (ADNet) of the left eye.

The left eye is defined as seen on the image; it is actually the person's right eye (physically).

6.7.2.7 leftEyeCorners

const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::leftEyeCorners {60,64}

Landmark indices (ADNet) of the left eyes' corners.

6.7.2.8 mouthInner

const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::mouthInner {88,89,90,91,92,93,94,95}

Landmark indices (ADNet) on the mouth's inner lip borders.

6.7.2.9 mouthOuter

const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::mouthOuter {76,77,78,79,80,81,82,83,84,85,86,87}

Landmark indices (ADNet) on the mouth's outer contour.

6.7.2.10 nosetip

const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::nosetip {54}

Landmark index (ADNet) of the nose tip.

6.7.2.11 pairsInnerLip

const LandmarkIdPairs OFIQ_LIB::modules::landmarks::adnet::pairsInnerLip

Initial value:
```
{
        {89, 95},
        {90, 94},
        {91, 93}
    }
```

Landmark index pairs (ADNet) of inner lip pairs.

Useful to measure closedness of mouth.

6.7.2.12 pairsLeftEye

const LandmarkIdPairs OFIQ_LIB::modules::landmarks::adnet::pairsLeftEye

Initial value:
```
{
        {61, 67},
        {62, 66},
        {63, 65}
    }
```

Pair indices of landmarks (ADNet) for the left eye.

Useful to measure eye openess.

6.7.2.13 pairsMouthCenter

const LandmarkIdPairs OFIQ_LIB::modules::landmarks::adnet::pairsMouthCenter

Initial value:
```
{
      {90, 94}
    }
```

Landmark index pair (ADNet) of the inner mouth (lips) center.

Useful to measure closedness of mouth.

6.7.2.14 pairsRightEye

const LandmarkIdPairs OFIQ_LIB::modules::landmarks::adnet::pairsRightEye

Initial value:
```
{
      {69, 75},
      {70, 74},
      {71, 73}
    }
```

Landmark index pairs (ADNet) of landmarks for the right eye.

Useful to measure eye openess.

6.7.2.15 rightEye

const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::rightEye {68,69,70,71,72,73,74,75}

Landmark indices (ADNet) of the right eye.

The right eye is defined as seen on the image; it is actually the person's left eye (physically).

6.7.2.16 rightEyeCorners

const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::rightEyeCorners {68,72}

Landmark indices (ADNet) of the right eyes' corners.

6.8 OFIQ_LIB::modules::measures Namespace Reference

Provides measures implemented in OFIQ.

Classes

- class BackgroundUniformity

    Implementation of the background uniformity measure.

- class CompressionArtifacts

    Implementation of the no compression artifacts measure.

- class CropOfTheFaceImage

    Implementation of the crop of the face image measure.

- class DynamicRange

    Implementation of the dynamic range measure.

- class Executor

    This class takes care of the computation of the measures activated.

- class ExpressionNeutrality

    Provides a class implementing the expression neutrality measure.

- class EyesOpen

    Implementation of the eyes open measure.

- class EyesVisible

    Implementation of the eyes visible measure.

- class FaceOcclusionPrevention

    Implementation of the face occlusion prevention measure.

- class HeadPose

    Implementation of head pose measures.

- class HeadSize

    Implementation of the head size measure.

- class IlluminationUniformity

    Implementation of the illumination uniformity measure.

- class InterEyeDistance

    Implementation of the inter-eye distance measure.

- class Luminance

    Implementation of two luminance measures.

- class Measure

    Base class for measures implemented in OFIQ.

- class MeasureFactory

    Measure factor class.

- class MouthClosed

    Implementation of the mouth closed measure.

- class MouthOcclusionPrevention

    Implementation of the mouth occlusion prevention measure.

- class NaturalColour

    Implementation of the natural colour measure.

- class NoHeadCoverings

    Implementation of the no head covering measure.

- class OverExposurePrevention

    Implementation of the over-exposure prevention measure.

- class Sharpness

    Implemantation of the sharpness measure.

- struct SigmoidParameters

    Parameters of the sigmoid function based quality mapping.

- class SingleFacePresent

    Implementation of the single face present measure.

- class UnderExposurePrevention

    Implementation of the under-exposure prevention measure.

- class UnifiedQualityScore

    Implementation of the unified quality measure.

Functions

- void log (const std::string_view &msg)

    Logging function for writing debug messages to std::cout.

Variables

- static const bool execLogActive = false

    This variable enables logging to std::cout for debug purposes. By default the logging is switched off.

6.8.1 Detailed Description

Provides measures implemented in OFIQ.

6.8.2 Function Documentation

6.8.2.1 log()

```
void OFIQ_LIB::modules::measures::log (
            const std::string_view & msg )
```

Logging function for writing debug messages to std::cout.

Parameters

msg	Message to be logged.

6.8.3 Variable Documentation

6.8.3.1 execLogActive

```
const bool OFIQ_LIB::modules::measures::execLogActive = false  [static]
```

This variable enables logging to std::cout for debug purposes. By default the logging is switched off.

6.9 OFIQ_LIB::modules::poseEstimators Namespace Reference

Provides implementation of a head pose estimator.

Classes

- class HeadPose3DDFAV2

    Implementation of a head pose estimator.

6.9.1 Detailed Description

Provides implementation of a head pose estimator.

6.10 OFIQ_LIB::modules::segmentations Namespace Reference

Provides segmentation-related implementations.

Classes

- class FaceOcclusionSegmentation

  Class managing the separation of facial parts not occluded by non-facial parts from other parts.
- class FaceParsing

  Class managing the separation of facial parts not occluded by non-facial parts from other parts.

Enumerations

- enum class SegmentClassLabels {
  background , skin , l_brow , r_brow ,
  l_eye , r_eye , eye_g , l_ear ,
  r_ear , ear_r , nose , mouth ,
  u_lip , l_lip , neck , neck_l ,
  cloth , hair , hat , face }

  Enum class of the different face regioons that can be segmented.

6.10.1 Detailed Description

Provides segmentation-related implementations.

Namespace for implementations related to facial segmentations.

6.10.2 Enumeration Type Documentation

6.10.2.1 SegmentClassLabels

enum class OFIQ_LIB::modules::segmentations::SegmentClassLabels [strong]

Enum class of the different face regioons that can be segmented.

Enumerator

background	background label
skin	skin label
l_brow	left eye brow
r_brow	right eye brow
l_eye	left eye
r_eye	right eye brow

Enumerator

eye_g	eye glasses
l_ear	left ear
r_ear	right eye brow
ear_r	earring
nose	nose
mouth	mouth
u_lip	upper lip
l_lip	lower lip
neck	neck
neck_l	necklace
cloth	clothing
hair	hair
hat	head covering
face	face

Chapter 7

Class Documentation

7.1 OFIQ_LIB::modules::landmarks::ADNetFaceLandmarkExtractor Class Reference

Class implementing the FaceLandmarkExtractorInterface interface.

```
#include <adnet_landmarks.h>
```

Inheritance diagram for OFIQ_LIB::modules::landmarks::ADNetFaceLandmarkExtractor:

```
┌─────────────────────────────────────────────────────────┐
│        OFIQ_LIB::FaceLandmarkExtractorInterface          │
└─────────────────────────────────────────────────────────┘
                            ▲
                            │
┌─────────────────────────────────────────────────────────┐
│  OFIQ_LIB::modules::landmarks::ADNetFaceLandmarkExtractor │
└─────────────────────────────────────────────────────────┘
```
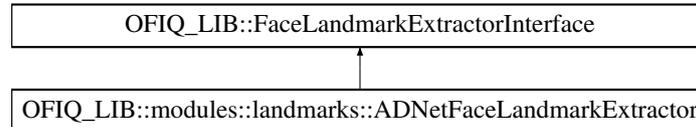
Public Member Functions

- ADNetFaceLandmarkExtractor (const Configuration &config)

  Constructor.
- ∼ADNetFaceLandmarkExtractor () override

  Destructor.

Public Member Functions inherited from **OFIQ_LIB::FaceLandmarkExtractorInterface**

- virtual ∼FaceLandmarkExtractorInterface ()=default

  Destructor.
- OFIQ::FaceLandmarks extractLandmarks (OFIQ_LIB::Session &session)

  Public method to extract landmarks from the image passed in the session object.

Protected Member Functions

- OFIQ::FaceLandmarks updateLandmarks (OFIQ_LIB::Session &session) override

  Computes landmarks of the face detected in the session.

Private Attributes

- std::unique_ptr< ADNetFaceLandmarkExtractorImpl > landmarkExtractor

    Encapsulated implementation class.

7.1.1 Detailed Description

Class implementing the FaceLandmarkExtractorInterface interface.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 ADNetFaceLandmarkExtractor()

```
OFIQ_LIB::modules::landmarks::ADNetFaceLandmarkExtractor::ADNetFaceLandmarkExtractor (
            const Configuration & config )  [explicit]
```

Constructor.

Parameters

config	Configuration object

7.1.2.2 ∼ADNetFaceLandmarkExtractor()

```
OFIQ_LIB::modules::landmarks::ADNetFaceLandmarkExtractor::∼ADNetFaceLandmarkExtractor ( )
[override]
```

Destructor.

7.1.3 Member Function Documentation

7.1.3.1 updateLandmarks()

```
OFIQ::FaceLandmarks OFIQ_LIB::modules::landmarks::ADNetFaceLandmarkExtractor::updateLandmarks
(
            OFIQ_LIB::Session & session )  [override], [protected], [virtual]
```

Computes landmarks of the face detected in the session.

The landmarks are computed using ADNet.

Parameters

session	Session object containing preprocessing results used by the function to compute the landmarks.

Returns

Facial landmarks.

Implements [OFIQ_LIB::FaceLandmarkExtractorInterface](#).

7.1.4 Member Data Documentation

7.1.4.1 landmarkExtractor

```
std::unique_ptr<ADNetFaceLandmarkExtractorImpl> OFIQ_LIB::modules::landmarks::ADNetFace←
LandmarkExtractor::landmarkExtractor  [private]
```

Encapsulated implementation class.

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

- [adnet_landmarks.h](#)

7.2 OFIQ_LIB::modules::measures::BackgroundUniformity Class Reference

Implementation of the background uniformity measure.

```
#include <BackgroundUniformity.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::BackgroundUniformity:

```
┌──────────────────────────────────────────────────┐
│      OFIQ_LIB::modules::measures::Measure          │
└──────────────────────────────────────────────────┘
                         ▲
┌──────────────────────────────────────────────────┐
│ OFIQ_LIB::modules::measures::BackgroundUniformity  │
└──────────────────────────────────────────────────┘
```
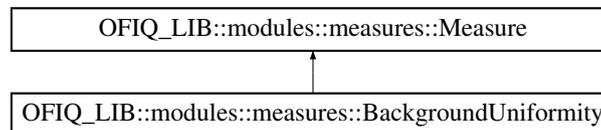
Public Member Functions

- [BackgroundUniformity](#) (const [Configuration](#) &[configuration](#), [Session](#) &session)

  Constructor.
- void [Execute](#) ([OFIQ_LIB::Session](#) &session) override

  Assesses uniformity of the background.

Public Member Functions inherited from [OFIQ_LIB::modules::measures::Measure](#)

- [Measure](#) (const [Configuration](#) &[configuration](#), const [OFIQ_LIB::Session](#) &, [OFIQ::QualityMeasure](#) measure)

  Constructor.
- virtual [~Measure](#) ()=default

  Destructor.
- virtual std::string [GetName](#) () const

  Returns the name of the measure.
- virtual [OFIQ::QualityMeasure](#) [GetQualityMeasure](#) () const

  Returns an enum encoding the measure.
- void [SetQualityMeasure](#) ([OFIQ_LIB::Session](#) &session, [OFIQ::QualityMeasure](#) measure, double rawValue, [OFIQ::QualityMeasureReturnCode](#) code)

  Inserts the result of a quality assessment in the session object.

Private Attributes

- uint16_t m_target_height = 292

  The aligned image and the face parsing mask is brought to an image of the target height before gradient computations and assessment is applied.

- uint16_t m_target_width = 354

  The aligned image and the face parsing mask is brought to an image of the target width before gradient computations and assessment is applied.

- uint16_t m_crop_left = 62

  Crop from the left side of the aligned image (returned by OFIQ_LIB::Session::getAlignedFace()) before scaling to the target dimension is applied.

- uint16_t m_crop_right = 62

  Crop from the right side of the aligned image (returned by OFIQ_LIB::Session::getAlignedFace()) before scaling to the target dimension is applied.

- uint16_t m_crop_top = 0

  Crop from the top of the aligned image (returned by OFIQ_LIB::Session::getAlignedFace()) before scaling to the target dimension is applied.

- uint16_t m_crop_bottom = 210

  Crop from the bottom of the aligned image (returned by OFIQ_LIB::Session::getAlignedFace()) before scaling to the target dimension is applied.

- uint16_t m_erosion_kernel_size = 4

  Size of the erosion kernel applied to the background as per OFIQ_LIB::modules::segmentations::FaceParsing to reduce the risk that background unformity estimation is applied to part of the subject.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

  Reads sigmoid-function based quality mapping from the configuration.

- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

  Reads sigmoid-function based quality mapping from the configuration.

- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

  Maps a native quality score to a quality component value.

- double ExecuteScalarConversion (const std::string &key, double rawValue)

  Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)

  Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- const Configuration & configuration

  Reference to the configuration with which the measure constructor has been invoked.

7.2.1 Detailed Description

Implementation of the background uniformity measure.

Uniformity of the backgound is measured on basis of the mean of the gradients computed on the background as per face parsing (see OFIQ_LIB::modules::segmentations::FaceParsing).

7.2.2 Constructor & Destructor Documentation

7.2.2.1 BackgroundUniformity()

```
OFIQ_LIB::modules::measures::BackgroundUniformity::BackgroundUniformity (
            const Configuration & configuration,
            Session & session )
```

Constructor.

Parameters

configuration	Configuration object from which the measure related configuration is read.
session	Session object containing the original facial image and pre-processing results, computed by the OFIQImpl::performPreprocessing().

7.2.3 Member Function Documentation

7.2.3.1 Execute()

```
void OFIQ_LIB::modules::measures::BackgroundUniformity::Execute (
            OFIQ_LIB::Session & session ) [override], [virtual]
```

Assesses uniformity of the background.

Assessment of the background uniformity is done by computing the mean of the background as per face parsing (see OFIQ_LIB::modules::segmentations::FaceParsing).

Parameters

session	Session object computed by the OFIQImpl::performPreprocessing() .

Implements OFIQ_LIB::modules::measures::Measure.

7.2.4 Member Data Documentation

7.2.4.1 m_crop_bottom

```
uint16_t OFIQ_LIB::modules::measures::BackgroundUniformity::m_crop_bottom = 210  [private]
```

Crop from the bottom of the aligned image (returned by OFIQ_LIB::Session::getAlignedFace()) before scaling to the target dimension is applied.

7.2.4.2 m_crop_left

```
uint16_t OFIQ_LIB::modules::measures::BackgroundUniformity::m_crop_left = 62  [private]
```

Crop from the left side of the aligned image (returned by OFIQ_LIB::Session::getAlignedFace()) before scaling to the target dimension is applied.

7.2.4.3 m_crop_right

```
uint16_t OFIQ_LIB::modules::measures::BackgroundUniformity::m_crop_right = 62  [private]
```

Crop from the right side of the aligned image (returned by OFIQ_LIB::Session::getAlignedFace()) before scaling to the target dimension is applied.

7.2.4.4 m_crop_top

```
uint16_t OFIQ_LIB::modules::measures::BackgroundUniformity::m_crop_top = 0  [private]
```

Crop from the top of the aligned image (returned by OFIQ_LIB::Session::getAlignedFace()) before scaling to the target dimension is applied.

7.2.4.5 m_erosion_kernel_size

```
uint16_t OFIQ_LIB::modules::measures::BackgroundUniformity::m_erosion_kernel_size = 4  [private]
```

Size of the erosion kernel applied to the background as per OFIQ_LIB::modules::segmentations::FaceParsing to reduce the risk that background unformity estimation is applied to part of the subject.

7.2.4.6 m_target_height

```
uint16_t OFIQ_LIB::modules::measures::BackgroundUniformity::m_target_height = 292  [private]
```

The aligned image and the face parsing mask is brought to an image of the target height before gradient computations and assessment is applied.

7.2.4.7 m_target_width

```
uint16_t OFIQ_LIB::modules::measures::BackgroundUniformity::m_target_width = 354  [private]
```

The aligned image and the face parsing mask is brought to an image of the target width before gradient computations and assessment is applied.

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

- BackgroundUniformity.h

7.3 OFIQ::BoundingBox Struct Reference

Data structure for descibing bounding boxes, e.g. the face region of the faces found by a face detector.

```
#include <ofiq_structs.h>
```

Public Member Functions

- BoundingBox ()=default

    Default constructor.
- BoundingBox (int16_t xleft, int16_t ytop, int16_t width, int16_t height, FaceDetectorType i_faceDetector)

    Parameterized constructor.

Public Attributes

- int16_t xleft { -1 }

    leftmost point on head, typically subject's right ear value must be on [0, imageWidth-1]
- int16_t ytop { -1 }

    high point of head, typically top of hair; value must be on [0, imageHeight-1]
- int16_t width { -1 }

    bounding box width
- int16_t height { -1 }

    bounding box height
- FaceDetectorType faceDetector = FaceDetectorType::NotSet

    Description of the face detector used.

7.3.1 Detailed Description

Data structure for descibing bounding boxes, e.g. the face region of the faces found by a face detector.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 BoundingBox() [1/2]

```
OFIQ::BoundingBox::BoundingBox ( )  [default]
```

Default constructor.

7.3.2.2 BoundingBox() [2/2]

```
OFIQ::BoundingBox::BoundingBox (
            int16_t xleft,
            int16_t ytop,
            int16_t width,
            int16_t height,
            FaceDetectorType i_faceDetector )  [inline]
```

Parameterized constructor.

Parameters

xleft	x coordinate of the upper left point of the bounding box.
ytop	y coordinate of the upper left point of the bounding box.
width	width of the bounding box.
height	height of the bounding box.
i_faceDetector	used face detector.

7.3.3 Member Data Documentation

7.3.3.1 faceDetector

FaceDetectorType OFIQ::BoundingBox::faceDetector = FaceDetectorType::NotSet

Description of the face detector used.

7.3.3.2 height

int16_t OFIQ::BoundingBox::height { -1 }

bounding box height

7.3.3.3 width

int16_t OFIQ::BoundingBox::width { -1 }

bounding box width

7.3.3.4 xleft

int16_t OFIQ::BoundingBox::xleft { -1 }

leftmost point on head, typically subject's right ear value must be on [0, imageWidth-1]

7.3.3.5 ytop

int16_t OFIQ::BoundingBox::ytop { -1 }

high point of head, typically top of hair; value must be on [0, imageHeight-1]

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

- ofiq_structs.h

7.4 OFIQ_LIB::modules::measures::CompressionArtifacts Class Reference

Implementation of the no compression artifacts measure.

```
#include <CompressionArtifacts.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::CompressionArtifacts:

```
┌─────────────────────────────────────────────────┐
│     OFIQ_LIB::modules::measures::Measure          │
└─────────────────────────────────────────────────┘
                        ▲
┌─────────────────────────────────────────────────┐
│ OFIQ_LIB::modules::measures::CompressionArtifacts │
└─────────────────────────────────────────────────┘
```
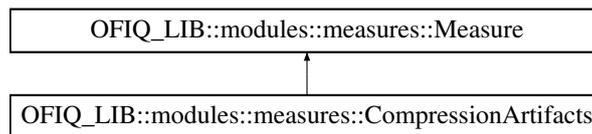
Public Member Functions

- CompressionArtifacts (const Configuration &configuration, Session &session)

    Constructor.
- void Execute (OFIQ_LIB::Session &session) override

    Assesses abscence of compression artifacts.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

    Constructor.
- virtual ∼Measure ()=default

    Destructor.
- virtual std::string GetName () const

    Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const

    Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)

    Inserts the result of a quality assessment in the session object.

Private Attributes

- uint16_t m_crop

    Top, right, left, and bottom margin by which the aligned image is cropped.
- uint16_t m_dim

    Target dimension of cropped image being scaled before input to the CNN .
- ONNXRuntimeSegmentation m_onnxRuntimeEnv

    Manages CNN estimations.

Additional Inherited Members

Protected Member Functions inherited from **OFIQ_LIB::modules::measures::Measure**

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

  Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

  Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

  Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)

  Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from **OFIQ_LIB::modules::measures::Measure**

- static double Sigmoid (double x, double x0, double w)

  Sigmoid function.

Protected Attributes inherited from **OFIQ_LIB::modules::measures::Measure**

- const Configuration & configuration

  Reference to the configuration with which the measure constructor has been invoked.

7.4.1 Detailed Description

Implementation of the no compression artifacts measure.

Assessment of the abscence of compression artifact (both JPEG and JPEG2000) based on a CNN trained by the OFIQ development team.

7.4.2 Constructor & Destructor Documentation

7.4.2.1 CompressionArtifacts()

```
OFIQ_LIB::modules::measures::CompressionArtifacts::CompressionArtifacts (
          const Configuration & configuration,
          Session & session )
```

Constructor.

The configuration parameter must contain the following entry:

- `params.measures.CompressionArtifacts.model_path`: Path to the CNN model file in ONNX format.

The following entries can (but do not need to) be configured.

- `params.measures.CompressionArtifacts.dim`: If configured, the value must be 248 which corresponds to the dimension of the CNN's input layer; other values will result in an error being thrown when OFIQ's CNN is invoked.

- `params.measures.CompressionArtifacts.crop`: Top, right, bottom, and left margin by which the aligned input image will be cropped before being scaled to the target dimension input of the CNN.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

Exceptions

OFIQ_LIB::OFIQError	if no valid model path is configured.

7.4.3 Member Function Documentation

7.4.3.1 Execute()

```
void OFIQ_LIB::modules::measures::CompressionArtifacts::Execute (
            OFIQ_LIB::Session & session )  [override], [virtual]
```

Assesses abscence of compression artifacts.

Assessment of the abscence of compression artifact (both JPEG and JPEG2000) based on a CNN trained by the OFIQ development team.

Parameters

session	Session object computed by the OFIQImpl::performPreprocessing() method.

Implements OFIQ_LIB::modules::measures::Measure.

7.4.4 Member Data Documentation

7.4.4.1 m_crop

```
uint16_t OFIQ_LIB::modules::measures::CompressionArtifacts::m_crop  [private]
```

Top, right, left, and bottom margin by which the aligned image is cropped.

The value can be configured by passing a corresponding configuration to the constructor.

Warning

The value should be 184 such that an aligned input image of dimension 616 x 616 is cropped to an image of dimension 248 x 248.

7.4.4.2 m_dim

```
uint16_t OFIQ_LIB::modules::measures::CompressionArtifacts::m_dim [private]
```

Target dimension of cropped image being scaled before input to the CNN .

The cropped image is scaled to the dimension m_dim x m_dim. The value can be configured by passing a corresponding configuration to the constructor.

Warning

> The value should be 248; if configured differently, do not expect that the cropped image can be successfully be passed to the CNN.

7.4.4.3 m_onnxRuntimeEnv

```
ONNXRuntimeSegmentation OFIQ_LIB::modules::measures::CompressionArtifacts::m_onnxRuntimeEnv
[private]
```

Manages CNN estimations.

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

- CompressionArtifacts.h

7.5 OFIQ_LIB::Configuration Class Reference

Configuration class.

```
#include <Configuration.h>
```

Public Member Functions

- Configuration (const std::string &configDir, const std::string &configFilename)
  Constructor.
- bool GetBool (const std::string &key, bool &value) const
  Accesses a boolean configuration.
- bool GetString (const std::string &key, std::string &value) const
  Accesses a string configuration.
- bool GetNumber (const std::string &key, double &value) const
  Accesses a double configuration.
- bool GetStringList (const std::string &key, std::vector< std::string > &value) const
  Accesses an array of strings configured.
- bool GetBool (const std::string &key) const
  Accesses a boolean configuration.
- std::string GetString (const std::string &key) const
  Accesses a string configuration.
- double GetNumber (const std::string &key) const
  Accesses a double configuration.
- std::string getDataDir () const
  Access configuration directory.
- void SetDataDir (std::string dataDir)
  Sets the configuration directory.

Private Attributes

- std::map< std::string, tao::json::value, std::less<> > parameters

   Map holding all configuration that can be accessed using a string key.
- std::filesystem::path m_dataDir

   Path to the configuration directory.

7.5.1 Detailed Description

Configuration class.

The class consumes the `taoJSON` library. A configuration is read from a `JAXN`-formatted file.

7.5.2 Constructor & Destructor Documentation

7.5.2.1 Configuration()

```
OFIQ_LIB::Configuration::Configuration (
            const std::string & configDir,
            const std::string & configFilename )
```

Constructor.

Parameters

configDir	Directory from which a JAXN configuration is read. The path can be absolute or relative to the path of the current working directory.
configFilename	Name of the JAXN configuration file in `configDir`.

7.5.3 Member Function Documentation

7.5.3.1 GetBool() [1/2]

```
bool OFIQ_LIB::Configuration::GetBool (
            const std::string & key ) const
```

Accesses a boolean configuration.

Parameters

key	Key of the configuration.

Returns

   The accessed boolean configuration.

Exceptions

OFIQ_LIB::OFIQError	if the configuration was not successfully accessed.

7.5.3.2 GetBool() [2/2]

```
bool OFIQ_LIB::Configuration::GetBool (
            const std::string & key,
            bool & value ) const
```

Accesses a boolean configuration.

Parameters

key	Key of the configuration.
value	Boolean reference to where the configuration result is stored.

Returns

> `true` if the configuration was successfully accessed; otherwise, if the configuration was not successfully accessed, the function returns `false`.

7.5.3.3 getDataDir()

```
std::string OFIQ_LIB::Configuration::getDataDir ( ) const
```

Access configuration directory.

The configuration directory has been specified in the constructor or afterwards by the SetDataDir() method.

Returns

> String representation of the configuration directory.

7.5.3.4 GetNumber() [1/2]

```
double OFIQ_LIB::Configuration::GetNumber (
            const std::string & key ) const
```

Accesses a double configuration.

Parameters

key	Key of the configuration.

Returns

> The accessed double configuration.

Exceptions

OFIQ_LIB::OFIQError	if the configuration was not successfully accessed.

7.5.3.5 GetNumber() [2/2]

```
bool OFIQ_LIB::Configuration::GetNumber (
            const std::string & key,
            double & value ) const
```

Accesses a double configuration.

Parameters

key	Key of the configuration.
value	Reference to a double type to where the configuration result is stored.

Returns

> `true` if the configuration was successfully accessed; otherwise, if the configuration was not successfully accessed, the function returns `false`.

7.5.3.6 GetString() [1/2]

```
std::string OFIQ_LIB::Configuration::GetString (
            const std::string & key ) const
```

Accesses a string configuration.

Parameters

key	Key of the configuration.

Returns

> The accessed string configuration.

Exceptions

OFIQ_LIB::OFIQError	if the configuration was not successfully accessed.

7.5.3.7 GetString() [2/2]

```
bool OFIQ_LIB::Configuration::GetString (
            const std::string & key,
            std::string & value ) const
```

Accesses a string configuration.

Parameters

key	Key of the configuration.
value	std::string reference to where the configuration result is stored.

Returns

> `true` if the configuration was successfully accessed; otherwise, if the configuration was not successfully accessed, the function returns `false`.

7.5.3.8 GetStringList()

```
bool OFIQ_LIB::Configuration::GetStringList (
            const std::string & key,
            std::vector< std::string > & value ) const
```

Accesses an array of strings configured.

The string list will not be emptied; any strings read will be appended.

Parameters

key	Key of the configuration.
value	Reference to a string list to where the configuration result is stored.

Returns

> `true` if the configuration was successfully accessed; otherwise, if the configuration was not successfully accessed, the function returns `false`.

7.5.3.9 SetDataDir()

```
void OFIQ_LIB::Configuration::SetDataDir (
            std::string dataDir )
```

Sets the configuration directory.

Parameters

dataDir	String representation of the configuration directory.

Attention

The configurations will not be updated when this method is called; it causes only an update of an internal private path member.

7.5.4 Member Data Documentation

7.5.4.1 m_dataDir

```
std::filesystem::path OFIQ_LIB::Configuration::m_dataDir  [private]
```

Path to the configuration directory.

The member is set either by the constructor or by the SetDataDir() method.

7.5.4.2 parameters

```
std::map<std::string, tao::json::value, std::less<> > OFIQ_LIB::Configuration::parameters
[private]
```

Map holding all configuration that can be accessed using a string key.

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

- Configuration.h

7.6 OFIQ_LIB::modules::measures::CropOfTheFaceImage Class Reference

Implementation of the crop of the face image measure.

```
#include <CropOfTheFaceImage.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::CropOfTheFaceImage:

```
┌─────────────────────────────────────────────────────┐
│        OFIQ_LIB::modules::measures::Measure           │
└─────────────────────────────────────────────────────┘
                          ▲
                          │
┌─────────────────────────────────────────────────────┐
│  OFIQ_LIB::modules::measures::CropOfTheFaceImage      │
└─────────────────────────────────────────────────────┘
```
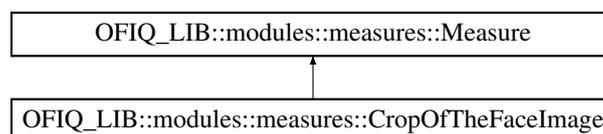
Public Member Functions

- CropOfTheFaceImage (const Configuration &configuration, Session &session)

  Constructor.
- void Execute (OFIQ_LIB::Session &session) override

  Implementation of the crop of the face image measure.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

    Constructor.
- virtual ~Measure ()=default

    Destructor.
- virtual std::string GetName () const

    Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const

    Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)

    Inserts the result of a quality assessment in the session object.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

    Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)

    Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)

    Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- const Configuration & configuration

    Reference to the configuration with which the measure constructor has been invoked.

7.6.1 Detailed Description

Implementation of the crop of the face image measure.

The crop of the face images measures whether the face is centered on the input image by comparing the resolution of the image with the landmarks detected during the pre-processing.

7.6.2 Constructor & Destructor Documentation

7.6.2.1 CropOfTheFaceImage()

```
OFIQ_LIB::modules::measures::CropOfTheFaceImage::CropOfTheFaceImage (
            const Configuration & configuration,
            Session & session )
```

Constructor.

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.6.3 Member Function Documentation

7.6.3.1 Execute()

```
void OFIQ_LIB::modules::measures::CropOfTheFaceImage::Execute (
            OFIQ_LIB::Session & session ) [override], [virtual]
```

Implementation of the crop of the face image measure.

The crop of the face images measures whether the face is centered on the input image by comparing the resolution of the image with the landmarks detected during the pre-processing.

Parameters

session	Session object.

Implements OFIQ_LIB::modules::measures::Measure.

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

- CropOfTheFaceImage.h

7.7 OFIQ_LIB::modules::measures::DynamicRange Class Reference

Implementation of the dynamic range measure.

```
#include <DynamicRange.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::DynamicRange:

```
┌─────────────────────────────────────────────────┐
│      OFIQ_LIB::modules::measures::Measure          │
└─────────────────────────────────────────────────┘
                        ▲
┌─────────────────────────────────────────────────┐
│    OFIQ_LIB::modules::measures::DynamicRange       │
└─────────────────────────────────────────────────┘
```
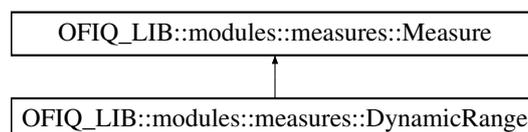
Public Member Functions

- DynamicRange (const Configuration &configuration, Session &session)
    Constructor.
- void Execute (OFIQ_LIB::Session &session) override
    Assesses dynamic range.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

  Constructor.
- virtual ∼Measure ()=default

  Destructor.
- virtual std::string GetName () const

  Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const

  Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)

  Inserts the result of a quality assessment in the session object.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

  Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

  Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

  Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)

  Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)

  Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- const Configuration & configuration

  Reference to the configuration with which the measure constructor has been invoked.

7.7.1 Detailed Description

Implementation of the dynamic range measure.

The dynamic range is computed from the luminance histogram. of the facial image.

7.7.2 Constructor & Destructor Documentation

7.7.2.1 DynamicRange()

```
OFIQ_LIB::modules::measures::DynamicRange::DynamicRange (
        const Configuration & configuration,
        Session & session )
```

Constructor.

Parameters

configuration	Configuration object from which the measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the [OFIQImpl::performPreprocessing()](#) method

7.7.3 Member Function Documentation

7.7.3.1 Execute()

```
void OFIQ_LIB::modules::measures::DynamicRange::Execute (
            OFIQ_LIB::Session & session ) [override], [virtual]
```

Assesses dynamic range.

Assessment of the dynamic range is computed from the luminance histogram.

Parameters

session	Session object.

Implements [OFIQ_LIB::modules::measures::Measure](#).

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

- [DynamicRange.h](#)

7.8 OFIQ_LIB::modules::measures::Executor Class Reference

This class takes care of the computation of the measures activated.

```
#include <Executor.h>
```

Public Member Functions

- [Executor](#) (std::vector< std::unique_ptr< [Measure](#) > > [measures](#))

    Construct a new Executor object.
- void [ExecuteAll](#) ([Session](#) &i_currentSession) const

    Run the computation of the activated measures on the data of the provided session.
- const std::vector< std::unique_ptr< [Measure](#) > > & [GetMeasures](#) () const

    Return the list of the activated measures.

Private Attributes

- std::vector< std::unique_ptr< [Measure](#) > > [measures](#)

    Container for access to the measures to be computed.

7.8.1 Detailed Description

This class takes care of the computation of the measures activated.

7.8.2 Constructor & Destructor Documentation

7.8.2.1 Executor()

```
OFIQ_LIB::modules::measures::Executor::Executor (
            std::vector< std::unique_ptr< Measure > > measures )  [inline], [explicit]
```

Construct a new Executor object.

Parameters

measures	Provide access to the activated measures.

7.8.3 Member Function Documentation

7.8.3.1 ExecuteAll()

```
void OFIQ_LIB::modules::measures::Executor::ExecuteAll (
            Session & i_currentSession ) const
```

Run the computation of the activated measures on the data of the provided session.

Parameters

i_currentSession	Container providing the data required for the computation of the measures.

7.8.3.2 GetMeasures()

```
const std::vector< std::unique_ptr< Measure > > & OFIQ_LIB::modules::measures::Executor::↩
GetMeasures ( ) const  [inline]
```

Return the list of the activated measures.

7.8.4 Member Data Documentation

7.8.4.1 measures

```
std::vector<std::unique_ptr<Measure> > OFIQ_LIB::modules::measures::Executor::measures  [private]
```

Container for access to the measures to be computed.

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

- Executor.h

7.9 OFIQ_LIB::modules::measures::ExpressionNeutrality Class Reference

Provides a class implementing the expression neutrality measure.

```
#include <ExpressionNeutrality.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::ExpressionNeutrality:

```
┌─────────────────────────────────────────────────────┐
│      OFIQ_LIB::modules::measures::Measure            │
└─────────────────────────────────────────────────────┘
                        ▲
                        │
┌─────────────────────────────────────────────────────┐
│  OFIQ_LIB::modules::measures::ExpressionNeutrality   │
└─────────────────────────────────────────────────────┘
```
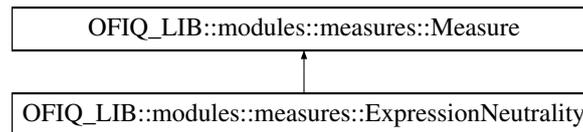
Public Member Functions

- ExpressionNeutrality (const Configuration &configuration, Session &session)

    Construct a new Expression Neutrality object.
- void Execute (OFIQ_LIB::Session &session) override

    Run the computation based on the data passed by the session object.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

    Constructor.
- virtual ∼Measure ()=default

    Destructor.
- virtual std::string GetName () const

    Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const

    Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)

    Inserts the result of a quality assessment in the session object.

Private Attributes

- ONNXRuntimeSegmentation m_onnxRuntimeEnvCNN1

    Instance of the enet_b0_8_best_vgaf_embed2 model. Set by ExpressionNeutrality.cnn1_model_path in the configuration file.
- ONNXRuntimeSegmentation m_onnxRuntimeEnvCNN2

    Instance of the enet_b2_8 model. Set by ExpressionNeutrality.cnn2_model_path in the configuration file.
- std::shared_ptr< cv::ml::Boost > classifier

    Instance of the AdaBoost classifier Set by ExpressionNeutrality.adaboost_model_path in the configuration file.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

    Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)

    Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)

    Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- const Configuration & configuration

    Reference to the configuration with which the measure constructor has been invoked.

7.9.1 Detailed Description

Provides a class implementing the expression neutrality measure.

The algorithm uses the CNN models enet_b0_8_best_vgaf and enet_b2_8 from `https://github.com/`↩ `HSE-asavchenko/face-emotion-recognition` as feature extractors and an AdaBoost classifier implemented in OpenCV. Inspired by Grimmer et al. [9], both CNN models have been modified to also output the embeddings of the second last layer, and have been converted to ONNX.

7.9.2 Constructor & Destructor Documentation

7.9.2.1 ExpressionNeutrality()

```
OFIQ_LIB::modules::measures::ExpressionNeutrality::ExpressionNeutrality (
            const Configuration & configuration,
            Session & session )
```

Construct a new Expression Neutrality object.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.9.3 Member Function Documentation

7.9.3.1 Execute()

```
void OFIQ_LIB::modules::measures::ExpressionNeutrality::Execute (
            OFIQ_LIB::Session & session )  [override], [virtual]
```

Run the computation based on the data passed by the session object.

Parameters

session	Session object

Implements OFIQ_LIB::modules::measures::Measure.

7.9.4 Member Data Documentation

7.9.4.1 classifier

```
std::shared_ptr<cv::ml::Boost> OFIQ_LIB::modules::measures::ExpressionNeutrality::classifier
[private]
```

Instance of the AdaBoost classifier Set by ExpressionNeutrality.adaboost_model_path in the configuration file.

7.9.4.2 m_onnxRuntimeEnvCNN1

```
ONNXRuntimeSegmentation OFIQ_LIB::modules::measures::ExpressionNeutrality::m_onnxRuntimeEnv←
CNN1  [private]
```

Instance of the enet_b0_8_best_vgaf_embed2 model. Set by ExpressionNeutrality.cnn1_model_path in the configuration file.

7.9.4.3 m_onnxRuntimeEnvCNN2

```
ONNXRuntimeSegmentation OFIQ_LIB::modules::measures::ExpressionNeutrality::m_onnxRuntimeEnv←
CNN2  [private]
```

Instance of the enet_b2_8 model. Set by ExpressionNeutrality.cnn2_model_path in the configuration file.

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

- ExpressionNeutrality.h

7.10 OFIQ_LIB::modules::measures::EyesOpen Class Reference

Implementation of the eyes open measure.

```
#include <EyesOpen.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::EyesOpen:

```
┌─────────────────────────────────────────────┐
│ OFIQ_LIB::modules::measures::Measure          │
└─────────────────────────────────────────────┘
                      ▲
┌─────────────────────────────────────────────┐
│ OFIQ_LIB::modules::measures::EyesOpen         │
└─────────────────────────────────────────────┘
```
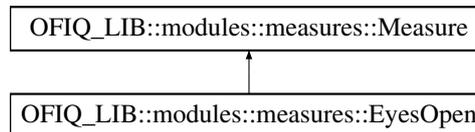
Public Member Functions

- EyesOpen (const Configuration &configuration, Session &session)

    Constructor.
- void Execute (OFIQ_LIB::Session &session) override

    Assesses eyes openness.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

    Constructor.
- virtual ∼Measure ()=default

    Destructor.
- virtual std::string GetName () const

    Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const

    Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)

    Inserts the result of a quality assessment in the session object.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

    Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)

    Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)

    Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- const Configuration & configuration

    Reference to the configuration with which the measure constructor has been invoked.

7.10.1   Detailed Description

Implementation of the eyes open measure.

Eyes openness is based on computing eyes aspect ratios of both eyes from eye landmarks.

7.10.2   Constructor & Destructor Documentation

7.10.2.1   EyesOpen()

```
OFIQ_LIB::modules::measures::EyesOpen::EyesOpen (
            const Configuration & configuration,
            Session & session )
```

Constructor.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.10.3   Member Function Documentation

7.10.3.1   Execute()

```
void OFIQ_LIB::modules::measures::EyesOpen::Execute (
            OFIQ_LIB::Session & session ) [override], [virtual]
```

Assesses eyes openness.

Eyes openness is based on computing eyes aspect ratios of both eyes from eye landmarks.

Parameters

session	Session object computed by the OFIQImpl::performPreprocessing() method.

See also

Session::getAlignedFaceLandmarks()

Implements OFIQ_LIB::modules::measures::Measure.

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

- EyesOpen.h

7.11 OFIQ_LIB::modules::measures::EyesVisible Class Reference

Implementation of the eyes visible measure.

```
#include <EyesVisible.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::EyesVisible:

```
┌─────────────────────────────────────────────┐
│  OFIQ_LIB::modules::measures::Measure         │
└─────────────────────────────────────────────┘
                      ▲
                      │
┌─────────────────────────────────────────────┐
│  OFIQ_LIB::modules::measures::EyesVisible     │
└─────────────────────────────────────────────┘
```
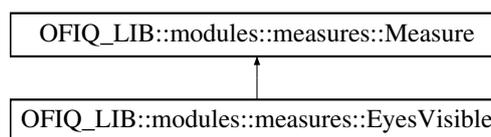
Public Member Functions

- EyesVisible (const Configuration &configuration, Session &session)

    Constructor.
- void Execute (OFIQ_LIB::Session &session) override

    Assesses eyes visibility.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

    Constructor.
- virtual ∼Measure ()=default

    Destructor.
- virtual std::string GetName () const

    Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const

    Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)

    Inserts the result of a quality assessment in the session object.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

    Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)

    Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)

    Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- const Configuration & configuration

    Reference to the configuration with which the measure constructor has been invoked.

7.11.1 Detailed Description

Implementation of the eyes visible measure.

Eyes visibility is assessed by measuring the coverage of the eye visibility zone with the result of face occlusion segmentation computed during pre-processing.

7.11.2 Constructor & Destructor Documentation

7.11.2.1 EyesVisible()

```
OFIQ_LIB::modules::measures::EyesVisible::EyesVisible (
            const Configuration & configuration,
            Session & session )
```

Constructor.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.11.3 Member Function Documentation

7.11.3.1 Execute()

```
void OFIQ_LIB::modules::measures::EyesVisible::Execute (
            OFIQ_LIB::Session & session ) [override], [virtual]
```

Assesses eyes visibility.

Eyes visibility is assessed by measuring the coverage of the eye visibility zone with the result of face occlusion segmentation computed during pre-processing by FaceOcclusionSegmentation. The pre-processing results are given by the session parameter.

Parameters

session	Session object computed by the OFIQImpl::performPreprocessing() method.

Implements OFIQ_LIB::modules::measures::Measure.

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

- EyesVisible.h

7.12 OFIQ_LIB::FaceDetectorInterface Class Reference

Provides the interface class to the face detector implementations.

```
#include <detectors.h>
```

Inheritance diagram for OFIQ_LIB::FaceDetectorInterface:
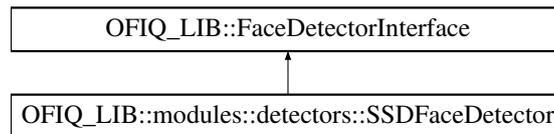
Public Member Functions

- virtual ∼FaceDetectorInterface ()=default

    Destroy the Face Detector Interface object.
- std::vector< OFIQ::BoundingBox > detectFaces (OFIQ_LIB::Session &session)

    This function detects faces in given image.

Protected Member Functions

- virtual std::vector< OFIQ::BoundingBox > UpdateFaces (OFIQ_LIB::Session &session)=0

    This method is to be called in derived classes to perform the detection of one/more faces on the given image.

7.12.1 Detailed Description

Provides the interface class to the face detector implementations.

This class provides the base class / interface for the integration of different implementations of a face detector.

7.12.2 Constructor & Destructor Documentation

7.12.2.1 ∼FaceDetectorInterface()

```
virtual OFIQ_LIB::FaceDetectorInterface::∼FaceDetectorInterface ( )  [virtual], [default]
```

Destroy the Face Detector Interface object.

7.12.3 Member Function Documentation

7.12.3.1 detectFaces()

```
std::vector< OFIQ::BoundingBox > OFIQ_LIB::FaceDetectorInterface::detectFaces (
            OFIQ_LIB::Session & session )
```

This function detects faces in given image.

Parameters

in	*session*	Session containing relevant information for the current task.

7.12.3.2 UpdateFaces()

```
virtual std::vector< OFIQ::BoundingBox > OFIQ_LIB::FaceDetectorInterface::UpdateFaces (
            OFIQ_LIB::Session & session )  [protected], [pure virtual]
```

This method is to be called in derived classes to perform the detection of one/more faces on the given image.

Parameters

session	Session containing relevant information for the current task.

Returns

> std::vector<OFIQ::BoundingBox>

Implemented in OFIQ_LIB::modules::detectors::SSDFaceDetector.

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

- detectors.h

7.13 OFIQ::FaceImageQualityAssessment Struct Reference

Data structure storing the results of the different measurement computations.

```
#include <ofiq_structs.h>
```

Public Member Functions

- FaceImageQualityAssessment ()=default

    Default contructor.
- FaceImageQualityAssessment (const QualityAssessments &qAssessments, const BoundingBox &boundingBox)

    Parameterized constructor.

Public Attributes

- QualityAssessments qAssessments

    Container for storing the resuls of the different measure computations.
- BoundingBox boundingBox

    Face region described by bounding box.

7.13.1 Detailed Description

Data structure storing the results of the different measurement computations.

7.13.2 Constructor & Destructor Documentation

7.13.2.1 FaceImageQualityAssessment() [1/2]

```
OFIQ::FaceImageQualityAssessment::FaceImageQualityAssessment ( )  [default]
```

Default contructor.

7.13.2.2 FaceImageQualityAssessment() [2/2]

```
OFIQ::FaceImageQualityAssessment::FaceImageQualityAssessment (
            const QualityAssessments & qAssessments,
            const BoundingBox & boundingBox )  [inline]
```

Parameterized constructor.

Parameters

in	*qAssessments*	
in	*boundingBox*	

7.13.3 Member Data Documentation

7.13.3.1 boundingBox

BoundingBox OFIQ::FaceImageQualityAssessment::boundingBox

Face region described by bounding box.

7.13.3.2 qAssessments

QualityAssessments OFIQ::FaceImageQualityAssessment::qAssessments

Container for storing the resuls of the different measure computations.

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

- ofiq_structs.h

7.14 OFIQ_LIB::FaceLandmarkExtractorInterface Class Reference

Implements the base class for the face landmark extractors.

```
#include <landmarks.h>
```

Inheritance diagram for OFIQ_LIB::FaceLandmarkExtractorInterface:
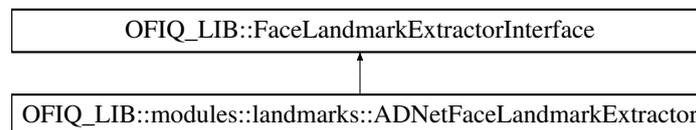
Public Member Functions

- virtual ∼FaceLandmarkExtractorInterface ()=default

  Destructor.
- OFIQ::FaceLandmarks extractLandmarks (OFIQ_LIB::Session &session)

  Public method to extract landmarks from the image passed in the session object.

Protected Member Functions

- virtual OFIQ::FaceLandmarks updateLandmarks (OFIQ_LIB::Session &session)=0

  Internal implementation of the derived class for extracting landmarks.

7.14.1 Detailed Description

Implements the base class for the face landmark extractors.

7.14.2 Constructor & Destructor Documentation

7.14.2.1 ∼**FaceLandmarkExtractorInterface()**

```
virtual OFIQ_LIB::FaceLandmarkExtractorInterface::∼FaceLandmarkExtractorInterface ( ) [virtual],
[default]
```

Destructor.

7.14.3 Member Function Documentation

7.14.3.1 extractLandmarks()

```
OFIQ::FaceLandmarks OFIQ_LIB::FaceLandmarkExtractorInterface::extractLandmarks (
            OFIQ_LIB::Session & session )
```

Public method to extract landmarks from the image passed in the session object.

Parameters

session	Data container, including the original image and preprocessed data.

Returns

OFIQ::FaceLandmarks

7.14.3.2 updateLandmarks()

```
virtual OFIQ::FaceLandmarks OFIQ_LIB::FaceLandmarkExtractorInterface::updateLandmarks (
            OFIQ_LIB::Session & session )  [protected], [pure virtual]
```

Internal implementation of the derived class for extracting landmarks.

Parameters

session	Data container, including the original image and preprocessed data.

Returns

OFIQ::FaceLandmarks

Implemented in OFIQ_LIB::modules::landmarks::ADNetFaceLandmarkExtractor.

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

- landmarks.h

7.15 OFIQ::FaceLandmarks Struct Reference

Data structure for storing facial landmarks.

```
#include <ofiq_structs.h>
```

Public Member Functions

- FaceLandmarks ()=default

Public Attributes

- LandmarkType type { LandmarkType::NotSet }
- Landmarks landmarks

7.15.1 Detailed Description

Data structure for storing facial landmarks.

7.15.2 Constructor & Destructor Documentation

7.15.2.1 FaceLandmarks()

```
OFIQ::FaceLandmarks::FaceLandmarks ( ) [default]
```

Default constructor.

7.15.3 Member Data Documentation

7.15.3.1 landmarks

```
Landmarks OFIQ::FaceLandmarks::landmarks
```

container for all detected landmarks.

7.15.3.2 type

```
LandmarkType OFIQ::FaceLandmarks::type { LandmarkType::NotSet }
```

Enum describing the type of the landmarks.

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

- ofiq_structs.h

7.16 OFIQ_LIB::modules::landmarks::FaceMeasures Class Reference

Provides static functions doing computations with landmarks.

```
#include <FaceMeasures.h>
```

Public Member Functions

- FaceMeasures ()=delete

    Constructor is deleted to avoid instantiations of this class.

Static Public Member Functions

- static double InterEyeDistance (const OFIQ::FaceLandmarks &faceLandmarks, double yaw)

    Computes the inter-eye distance based on the specified facial landmarks and yaw angle.
- static cv::Mat GetFaceMask (const OFIQ::FaceLandmarks &faceLandmarks, const int height, const int width, const float alpha=0)

    Creates a binary image of specified dimension and masks all pixels inside or on the convex hull.
- static double GetDistance (const OFIQ::LandmarkPoint &a, const OFIQ::LandmarkPoint &b)

    Convenience method for computing the Euclidean distance between two landmark points.
- static double GetDistance (const LandmarkPair &pair)

    Convenience method computing the Euclidean distance between two landmark points.
- static OFIQ::LandmarkPoint GetMiddle (const OFIQ::Landmarks &landmarks)

    Computes the center point of the specified landmarks.
- static OFIQ::LandmarkPoint GetMiddle (const LandmarkPair &pair)

    Computes the point in between two landmark points.
- static OFIQ::LandmarkPoint GetMiddle (const std::vector< LandmarkPair > &pairs)

    Computes the center of the specified landmark points.
- static double GetMaxPairDistance (const OFIQ::FaceLandmarks &landmarks, landmarks::FaceParts face↩
Part)

    Returns this maximum of face pairs from landmarks corresponding to the specified face part.

7.16.1 Detailed Description

Provides static functions doing computations with landmarks.

7.16.2 Constructor & Destructor Documentation

7.16.2.1 FaceMeasures()

```
OFIQ_LIB::modules::landmarks::FaceMeasures::FaceMeasures ( ) [delete]
```

Constructor is deleted to avoid instantiations of this class.

7.16.3 Member Function Documentation

7.16.3.1 GetDistance() [1/2]

```
static double OFIQ_LIB::modules::landmarks::FaceMeasures::GetDistance (
            const LandmarkPair & pair ) [inline], [static]
```

Convenience method computing the Euclidean distance between two landmark points.

Parameters

pair	The two landmark points stored in the member `pair.Lower` and `pair.Upper`.

Returns

> Euclidean distance.

7.16.3.2 GetDistance() [2/2]

```
static double OFIQ_LIB::modules::landmarks::FaceMeasures::GetDistance (
            const OFIQ::LandmarkPoint & a,
            const OFIQ::LandmarkPoint & b )  [static]
```

Convenience method for computing the Euclidean distance between two landmark points.

Parameters

a	First landmark point
b	Second landmark point

Returns

> Euclidean distance between `a` and `b`.

7.16.3.3 GetFaceMask()

```
static cv::Mat OFIQ_LIB::modules::landmarks::FaceMeasures::GetFaceMask (
            const OFIQ::FaceLandmarks & faceLandmarks,
            const int height,
            const int width,
            const float alpha = 0 )  [static]
```

Creates a binary image of specified dimension and masks all pixels inside or on the convex hull.

All pixels on or inside the convex hull of the landmarks are set to 1; all other pixels are set to 0.

Parameters

faceLandmarks	Facial landmarks object
height	Height of the mask image
width	Width of the mask image
alpha	Should be 0; different values have only be used for NIST submissions.

Returns

> Mask image

7.16.3.4 GetMaxPairDistance()

```
static double OFIQ_LIB::modules::landmarks::FaceMeasures::GetMaxPairDistance (
            const OFIQ::FaceLandmarks & landmarks,
            landmarks::FaceParts facePart )  [static]
```

Returns this maximum of face pairs from landmarks corresponding to the specified face part.

Face parts (such as mouth) consist of landmarks that have a mate. For example, the lower lip point may correspond to the upper lip point. For such face parts, the maximal separation (e.g., useful for detecting the mouth closeness or eyes openness) can be computed.

Parameters

landmarks	Facial landmarks
facePart	Face part

Returns

Maxim of face pairs

7.16.3.5 GetMiddle() [1/3]

```
static OFIQ::LandmarkPoint OFIQ_LIB::modules::landmarks::FaceMeasures::GetMiddle (
            const LandmarkPair & pair )  [inline], [static]
```

Computes the point in between two landmark points.

Parameters

pair	Pair of landmark points.

Returns

Point between the two landmark points.

7.16.3.6 GetMiddle() [2/3]

```
static OFIQ::LandmarkPoint OFIQ_LIB::modules::landmarks::FaceMeasures::GetMiddle (
            const OFIQ::Landmarks & landmarks )  [static]
```

Computes the center point of the specified landmarks.

Parameters

landmarks	Facial landmarks

Returns

Center point of the landmarks.

7.16.3.7 GetMiddle() [3/3]

```
static OFIQ::LandmarkPoint OFIQ_LIB::modules::landmarks::FaceMeasures::GetMiddle (
            const std::vector< LandmarkPair > & pairs ) [inline], [static]
```

Computes the center of the specified landmark points.

This is a convenience method to compute the center if the landmarks are available as a vector of landmark pairs.

Parameters

pairs	List of landmark pairs.

Returns

Center of the landmark points.

7.16.3.8 InterEyeDistance()

```
static double OFIQ_LIB::modules::landmarks::FaceMeasures::InterEyeDistance (
            const OFIQ::FaceLandmarks & faceLandmarks,
            double yaw ) [static]
```

Computes the inter-eye distance based on the specified facial landmarks and yaw angle.

If (x_0, y_0) and (x_1, y_1) are the left and right eye centres, then the following inter-eye distance is computed.

$$\text{IED} = \|(x_0, y_0) - (x_1, y_1)\|_2 \cdot \frac{1}{\cos(\alpha)}$$

where

$$\alpha = \pi \cdot \text{yaw}/180$$

is the secant of the yaw angle.

Parameters

faceLandmarks	Facial landmarks
yaw	Yaw angle in degree

Returns

The inter-eye distance

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

- FaceMeasures.h

7.17 OFIQ_LIB::modules::measures::FaceOcclusionPrevention Class Reference

Implementation of the face occlusion prevention measure.

```
#include <FaceOcclusionPrevention.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::FaceOcclusionPrevention:

```
┌─────────────────────────────────────────────────────┐
│       OFIQ_LIB::modules::measures::Measure            │
└─────────────────────────────────────────────────────┘
                         ▲
                         │
┌─────────────────────────────────────────────────────┐
│  OFIQ_LIB::modules::measures::FaceOcclusionPrevention │
└─────────────────────────────────────────────────────┘
```
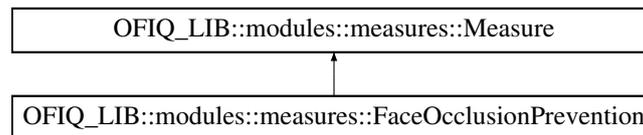
Public Member Functions

- FaceOcclusionPrevention (const Configuration &configuration, Session &session)

    Constructor.
- void Execute (OFIQ_LIB::Session &session) override

    Assesses absence of face occlusion.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

    Constructor.
- virtual ∼Measure ()=default

    Destructor.
- virtual std::string GetName () const

    Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const

    Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)

    Inserts the result of a quality assessment in the session object.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

    Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)

    Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)
    Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- const Configuration & configuration
    Reference to the configuration with which the measure constructor has been invoked.

7.17.1 Detailed Description

Implementation of the face occlusion prevention measure.

Absence of face occlusion is assessed by measuring the coverage of the marked region with the result of face occlusion segmentation computed during pre-processing.

7.17.2 Constructor & Destructor Documentation

7.17.2.1 FaceOcclusionPrevention()

```
OFIQ_LIB::modules::measures::FaceOcclusionPrevention::FaceOcclusionPrevention (
            const Configuration & configuration,
            Session & session )
```

Constructor.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.17.3 Member Function Documentation

7.17.3.1 Execute()

```
void OFIQ_LIB::modules::measures::FaceOcclusionPrevention::Execute (
            OFIQ_LIB::Session & session ) [override], [virtual]
```

Assesses absence of face occlusion.

Absence of face occlusion is assessed by measuring the coverage of the marked region with the result of face occlusion segmentation computed during pre-processing. Pre-processing results are passed to the method with the session parameter.

Parameters

session	Session object computed by the OFIQImpl::performPreprocessing() method.

See also

> FaceOcclusionSegmentation

Implements OFIQ_LIB::modules::measures::Measure.

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

- FaceOcclusionPrevention.h

7.18 OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation Class Reference

Class managing the separation of facial parts not occluded by non-facial parts from other parts.

```
#include <FaceOcclusionSegmentation.h>
```

Inheritance diagram for OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation:

OFIQ_LIB::SegmentationExtractorInterface

OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation
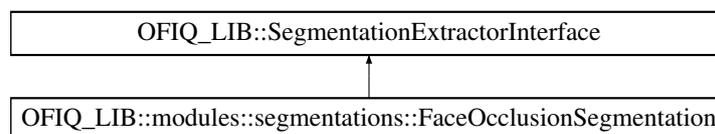
Public Member Functions

- FaceOcclusionSegmentation (const Configuration &config)

  Constructor.
- ∼FaceOcclusionSegmentation () override=default

  Destructor.

Public Member Functions inherited from OFIQ_LIB::SegmentationExtractorInterface

- virtual ∼SegmentationExtractorInterface ()=default

  Standard destructor.
- OFIQ::Image & GetMask (OFIQ_LIB::Session &session, modules::segmentations::SegmentClassLabels faceSegment)

  Get a mask of the face region requested.

Protected Member Functions

- OFIQ::Image UpdateMask (OFIQ_LIB::Session &session, modules::segmentations::SegmentClassLabels faceSegment) override

  Implements face occlusion segmentation.

Protected Member Functions inherited from OFIQ_LIB::SegmentationExtractorInterface

- std::string GetLastSessionId () const

  Accesses the last session id for this interface.

Private Member Functions

- cv::Mat GetFaceOcclusionSegmentation (const cv::Mat &alignedImage)

  Does the actual CNN-based occlusion-aware segmentation.

Private Attributes

- ONNXRuntimeSegmentation m_onnxRuntimeEnv

  Manages CNN computations.
- std::shared_ptr< cv::Mat > segmentationImage

  Stores the last result computed with UpdateMask().
- const std::string modelConfigItem = "params.measures.FaceOcclusionSegmentation.model_path"

  JSON/JAXN key to access path to FaceExtraction's model file from Configuration object.
- const int cropLeft = 96

  Cropping parameter.
- const int cropRight = 96

  Cropping parameter.
- const int cropTop = 96

  Cropping parameter.
- const int cropBottom = 96

  Cropping parameter.
- const int scaledWidth = 224

  After cropping the aligned image, the result will be scaled to a dimension of that width for being input to the CNN-based segmentation.
- const int scaledHeight = 224

  After cropping the aligned image, the result will be scaled to a dimension of that height for being input to the CNN-based segmentation.

7.18.1 Detailed Description

Class managing the separation of facial parts not occluded by non-facial parts from other parts.

The implementation is based on a CNN from `FaceExtraction`.

7.18.2 Constructor & Destructor Documentation

7.18.2.1 FaceOcclusionSegmentation()

```
OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation::FaceOcclusionSegmentation (
            const Configuration & config ) [explicit]
```

Constructor.

Parameters

config	Configuration object from which some segmentation-related parameters may be read.

See also

> Other required configurations

7.18.2.2 ∼**FaceOcclusionSegmentation()**

```
OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation::~FaceOcclusionSegmentation ( )
[override], [default]
```

Destructor.

7.18.3 Member Function Documentation

7.18.3.1 GetFaceOcclusionSegmentation()

```
cv::Mat OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation::GetFaceOcclusionSegmentation
(
            const cv::Mat & alignedImage )  [private]
```

Does the actual CNN-based occlusion-aware segmentation.

Parameters

alignedImage	Aligned image of dimension 616 x 616 as returned by Session::getAlignedFace().

Returns

> Image where a pixel belonging to non-occluded facial parts is encoded as the byte value 1 and pixels belonging to other parts are encoded by the byte value 0.

7.18.3.2 UpdateMask()

```
OFIQ::Image OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation::UpdateMask (
            OFIQ_LIB::Session & session,
            modules::segmentations::SegmentClassLabels faceSegment )  [override], [protected],
[virtual]
```

Implements face occlusion segmentation.

The function is invoked by SegmentationExtractorInterface::GetMask(). Invokes GetFaceOcclusionSegmentation() and stores its output in the private segmentationImage member.

Parameters

session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method.
faceSegment	Should be the value SegmentClassLabels::face.

Returns

    Face occlusion segmentation mask.

Implements OFIQ_LIB::SegmentationExtractorInterface.

7.18.4 Member Data Documentation

7.18.4.1 cropBottom

`const int OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation::cropBottom = 96 [private]`

Cropping parameter.

7.18.4.2 cropLeft

`const int OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation::cropLeft = 96 [private]`

Cropping parameter.

7.18.4.3 cropRight

`const int OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation::cropRight = 96 [private]`

Cropping parameter.

7.18.4.4 cropTop

`const int OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation::cropTop = 96 [private]`

Cropping parameter.

7.18.4.5 m_onnxRuntimeEnv

`ONNXRuntimeSegmentation OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation::m_onnx↩`
`RuntimeEnv [private]`

Manages CNN computations.

7.18.4.6 modelConfigItem

`const std::string OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation::modelConfigItem = "params.measures.FaceOcclusionSegmentation.model_path" [private]`

JSON/JAXN key to access path to FaceExtraction's model file from Configuration object.

7.18.4.7 scaledHeight

`const int OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation::scaledHeight = 224 [private]`

After cropping the aligned image, the result will be scaled to a dimension of that height for being input to the CNN-based segmentation.

7.18.4.8 scaledWidth

`const int OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation::scaledWidth = 224 [private]`

After cropping the aligned image, the result will be scaled to a dimension of that width for being input to the CNN-based segmentation.

7.18.4.9 segmentationImage

`std::shared_ptr<cv::Mat> OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation::segmentation↩Image [private]`

Stores the last result computed with UpdateMask().

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

- FaceOcclusionSegmentation.h

7.19 OFIQ_LIB::modules::segmentations::FaceParsing Class Reference

Class managing the separation of facial parts not occluded by non-facial parts from other parts.

`#include <FaceParsing.h>`

Inheritance diagram for OFIQ_LIB::modules::segmentations::FaceParsing:

OFIQ_LIB::SegmentationExtractorInterface

OFIQ_LIB::modules::segmentations::FaceParsing
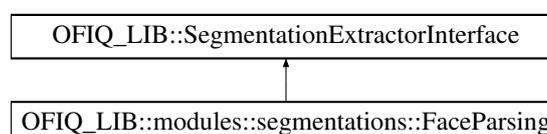
Public Member Functions

- FaceParsing (const Configuration &config)

  Constructor.
- ∼FaceParsing () override=default

  Destructor.

Public Member Functions inherited from OFIQ_LIB::SegmentationExtractorInterface

- virtual ∼SegmentationExtractorInterface ()=default

  Standard destructor.
- OFIQ::Image & GetMask (OFIQ_LIB::Session &session, modules::segmentations::SegmentClassLabels faceSegment)

  Get a mask of the face region requested.

Protected Member Functions

- OFIQ::Image UpdateMask (OFIQ_LIB::Session &session, modules::segmentations::SegmentClassLabels faceSegment) override

  Implements face parsing.

Protected Member Functions inherited from OFIQ_LIB::SegmentationExtractorInterface

- std::string GetLastSessionId () const

  Accesses the last session id for this interface.

Private Member Functions

- void SetImage (OFIQ_LIB::Session &session)

Static Private Member Functions

- static cv::Mat CreateBlob (const cv::Mat &image, int i_imageSize_one_dim)

  Creates the blob being input to the face parsing CNN.
- static std::shared_ptr< cv::Mat > CalculateClassIds (const cv::Mat &resultImage, int i_imageSize_one_dim)

  Applies segmentation to the blob created from the input image and returns the result.

Private Attributes

- ONNXRuntimeSegmentation m_onnxRuntimeEnv

  Manages CNN computations.
- std::shared_ptr< cv::Mat > segmentationImage

  Stores the last result computed with UpdateMask().
- const std::string modelConfigItem = "params.measures.FaceParsing.model_path"

  JSON/JAXN key to access path to `BiSeNet` *model in ONNX format from Configuration object.*
- const int imageSize = 400

  Face parsing target dimension.
- const int cropLeft = 30

  Cropping parameter.
- const int cropRight = 30

  Cropping parameter.
- const int cropTop = 0

  Cropping parameter.
- const int cropBottom = 60

  Cropping parameter.

7.19.1 Detailed Description

Class managing the separation of facial parts not occluded by non-facial parts from other parts.

Implements a BiSeNet-based face parsing. The aligned face image is cropped and then scaled to the dimension 400 x 400. All pixels of the resulting image are assigned to one of the following class.

value	class
0	background
1	face skin
2	left eye brow
3	right eye brow
4	left eye
5	right eye
6	eyeglasses
7	left ear
8	right ear
9	earring
10	nose
11	mouth
12	upper lip
13	lower lip
14	neck
15	necklace
16	clothing
17	hair
18	head covering

The result of face parsing is an image (matrix) of dimension 400 x 400 where each pixel is assigned with one of the values listed in the table from above.

7.19.2 Constructor & Destructor Documentation

7.19.2.1 FaceParsing()

```
OFIQ_LIB::modules::segmentations::FaceParsing::FaceParsing (
            const Configuration & config ) [explicit]
```

Constructor.

Parameters

config	Configuration object from which related parameters may be read.

See also

   For configuration of face parsing, see Other required configurations

7.19.2.2 ∼FaceParsing()

```
OFIQ_LIB::modules::segmentations::FaceParsing::∼FaceParsing ( ) [override], [default]
```

Destructor.

7.19.3 Member Function Documentation

7.19.3.1 CalculateClassIds()

```
static std::shared_ptr< cv::Mat > OFIQ_LIB::modules::segmentations::FaceParsing::Calculate←
ClassIds (
           const cv::Mat & resultImage,
           int i_imageSize_one_dim ) [static], [private]
```

Applies segmentation to the blob created from the input image and returns the result.

Is invoked by SetImage().

Parameters

resultImage	Blob being created by one of the CreateBlob functions.
i_imageSize_one_dim	Specifies the size of the blob being input to the face parsing CNN; should be 400, such that a blob of dimension 400 x 400 is created.

Returns

Result of face parsing.

7.19.3.2 CreateBlob()

```
static cv::Mat OFIQ_LIB::modules::segmentations::FaceParsing::CreateBlob (
           const cv::Mat & image,
           int i_imageSize_one_dim ) [static], [private]
```

Creates the blob being input to the face parsing CNN.

Parameters

image	Input image
i_imageSize_one_dim	Specifies the size of the blob being input to the face parsing CNN; should be 400, such that a blob of dimension 400 x 400 is created.

Returns

Blob of requested dimension.

7.19.3.3 SetImage()

```
void OFIQ_LIB::modules::segmentations::FaceParsing::SetImage (
            OFIQ_LIB::Session & session )  [private]
```

7.19.3.4 UpdateMask()

```
OFIQ::Image OFIQ_LIB::modules::segmentations::FaceParsing::UpdateMask (
            OFIQ_LIB::Session & session,
            modules::segmentations::SegmentClassLabels faceSegment )  [override], [protected],
[virtual]
```

Implements face parsing.

The function is invoked by SegmentationExtractorInterface::GetMask(). It crops the aligned face image returned by Session::getAlignedFace() as configured by private member variables. The result is scaled to the dimension of 400 x 400 and passed to the `BiSeNet` CNN. The output is returned as face parsing.

Parameters

session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method.
faceSegment	Enum value encoding the requested face segment. If the value is SegmentClassLabels::face then an image of dimension 400 x 400 is returned assigning each pixel a value between 0 and 18 as in the table of FaceParsing; otherwise a binary mask of dimension 400 x 400 is returned with the requested face segment and morphologically extended (kernel size 3) is returned.

Returns

Face parsing image of dimension 400 x 400 where each pixel is assigned with one of the values described in the table of the OFIQ_LIB::modules::segmentations::FaceParsing class documentation.

Implements OFIQ_LIB::SegmentationExtractorInterface.

7.19.4 Member Data Documentation

7.19.4.1 cropBottom

```
const int OFIQ_LIB::modules::segmentations::FaceParsing::cropBottom = 60  [private]
```

Cropping parameter.

7.19.4.2 cropLeft

```
const int OFIQ_LIB::modules::segmentations::FaceParsing::cropLeft = 30  [private]
```

Cropping parameter.

7.19.4.3 cropRight

const int OFIQ_LIB::modules::segmentations::FaceParsing::cropRight = 30 [private]

Cropping parameter.

7.19.4.4 cropTop

const int OFIQ_LIB::modules::segmentations::FaceParsing::cropTop = 0 [private]

Cropping parameter.

7.19.4.5 imageSize

const int OFIQ_LIB::modules::segmentations::FaceParsing::imageSize = 400 [private]

Face parsing target dimension.

7.19.4.6 m_onnxRuntimeEnv

ONNXRuntimeSegmentation OFIQ_LIB::modules::segmentations::FaceParsing::m_onnxRuntimeEnv [private]

Manages CNN computations.

7.19.4.7 modelConfigItem

const std::string OFIQ_LIB::modules::segmentations::FaceParsing::modelConfigItem = "params.↵
measures.FaceParsing.model_path" [private]

JSON/JAXN key to access path to BiSeNet model in ONNX format from Configuration object.

7.19.4.8 segmentationImage

std::shared_ptr<cv::Mat> OFIQ_LIB::modules::segmentations::FaceParsing::segmentationImage
[private]

Stores the last result computed with UpdateMask().

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

- FaceParsing.h

7.20 OFIQ_LIB::modules::measures::HeadPose Class Reference

Implementation of head pose measures.

```
#include <HeadPose.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::HeadPose:

```
┌──────────────────────────────────────────────┐
│  OFIQ_LIB::modules::measures::Measure          │
└──────────────────────────────────────────────┘
                      ▲
┌──────────────────────────────────────────────┐
│  OFIQ_LIB::modules::measures::HeadPose          │
└──────────────────────────────────────────────┘
```
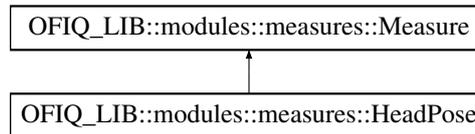
Public Member Functions

- HeadPose (const Configuration &configuration, Session &session)

    Constructor for HeadPose.
- void Execute (OFIQ_LIB::Session &session) override

    Assesses head pose measure.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

    Constructor.
- virtual ∼Measure ()=default

    Destructor.
- virtual std::string GetName () const

    Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const

    Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)

    Inserts the result of a quality assessment in the session object.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

    Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)

    Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)

    Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- const Configuration & configuration

    Reference to the configuration with which the measure constructor has been invoked.

7.20.1 Detailed Description

Implementation of head pose measures.

Head pose measures are returned for roll, pitch and yaw face angle.

7.20.2 Constructor & Destructor Documentation

7.20.2.1 HeadPose()

```
OFIQ_LIB::modules::measures::HeadPose::HeadPose (
            const Configuration & configuration,
            Session & session )
```

Constructor for HeadPose.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.20.3 Member Function Documentation

7.20.3.1 Execute()

```
void OFIQ_LIB::modules::measures::HeadPose::Execute (
            OFIQ_LIB::Session & session )  [override], [virtual]
```

Assesses head pose measure.

Quality components are computed with the help of the cosine of the native quality scores (angles).

Parameters

session	Session object computed by the OFIQImpl::performPreprocessing() method.

Implements OFIQ_LIB::modules::measures::Measure.

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

- HeadPose.h

7.21 OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2 Class Reference

Implementation of a head pose estimator.

```
#include <HeadPose3DDFAV2.h>
```

Inheritance diagram for OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2:

```
┌─────────────────────────────────────────────────────────────┐
│            OFIQ_LIB::PoseEstimatorInterface                  │
└─────────────────────────────────────────────────────────────┘
                              ▲
┌─────────────────────────────────────────────────────────────┐
│   OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2         │
└─────────────────────────────────────────────────────────────┘
```
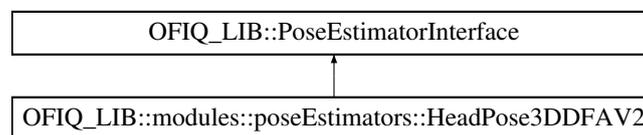
Public Member Functions

- HeadPose3DDFAV2 (const Configuration &config)

    Constructor for HeadPose3DDFAV2.
- ∼HeadPose3DDFAV2 () override=default

    Destructor.

Public Member Functions inherited from **OFIQ_LIB::PoseEstimatorInterface**

- virtual ∼PoseEstimatorInterface ()=default

    Standard destructor.
- EulerAngle & estimatePose (OFIQ_LIB::Session &session)

    This function estimates the three head orientation angles.

Protected Member Functions

- void updatePose (OFIQ_LIB::Session &session, EulerAngle &pose) override

    Computation of the head pose.

Private Member Functions

- cv::Mat CropImage (const cv::Mat &image, const OFIQ::BoundingBox &biggestFace)

    Crop face from image. Internally the passed bounding box will be transformed to a square region.

Private Attributes

- Ort::Env m_ortenv

  ONNXRuntime environment handle.
- std::unique_ptr< Ort::Session > m_ort_session

  ONNXRuntime session handle.
- int64_t m_expected_image_width = 0

  Width of the CNN used for computation, read from the loaded model.
- int64_t m_expected_image_height = 0

  Height of the CNN used for computation, read from the loaded model.
- int64_t m_expected_image_number_of_channels = 0

  Expected number of channels of the input image, read from the loaded model.
- int64_t m_number_of_input_elements = 0

  Number of input elements of the CNN used for computation, read from the loaded model.
- std::array< int64_t, 4 > inputShape

  inputShape of the CNN used for computation, read from the loaded model.

Static Private Attributes

- static const std::string paramPoseEstimatorModel

  Name of the used CNN net, passed from the configuration.

Additional Inherited Members

Public Types inherited from OFIQ_LIB::PoseEstimatorInterface

- using EulerAngle = std::array<double, 3>

7.21.1 Detailed Description

Implementation of a head pose estimator.

The estimator is is based on a CNN from https://github.com/cleardusk/3DDFA_V2.

7.21.2 Constructor & Destructor Documentation

7.21.2.1 HeadPose3DDFAV2()

```
OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::HeadPose3DDFAV2 (
        const Configuration & config )  [explicit]
```

Constructor for HeadPose3DDFAV2.

Parameters

config	Configuration from where the the path to the CNN model in ONNX format to read.

7.21.2.2 ∼HeadPose3DDFAV2()

```
OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::∼HeadPose3DDFAV2 ( ) [override], [default]
```

Destructor.

7.21.3 Member Function Documentation

7.21.3.1 CropImage()

```
cv::Mat OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::CropImage (
            const cv::Mat & image,
            const OFIQ::BoundingBox & biggestFace ) [private]
```

Crop face from image. Internally the passed bounding box will be transformed to a square region.

Parameters

image	Input image.
biggestFace	Input region to be cropped.

Returns

cv::Mat Cropped face region.

7.21.3.2 updatePose()

```
void OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::updatePose (
            OFIQ_LIB::Session & session,
            EulerAngle & pose ) [override], [protected], [virtual]
```

Computation of the head pose.

Parameters

session	Session object containing the original facial image and pre-processing results computed.
pose	Estimated head pose.

Implements OFIQ_LIB::PoseEstimatorInterface.

7.21.4 Member Data Documentation

7.21.4.1 inputShape

```
std::array<int64_t, 4> OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::inputShape [private]
```

inputShape of the CNN used for computation, read from the loaded model.

7.21.4.2 m_expected_image_height

`int64_t OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::m_expected_image_height = 0 [private]`

Height of the CNN used for computation, read from the loaded model.

7.21.4.3 m_expected_image_number_of_channels

`int64_t OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::m_expected_image_number_of_↩ channels = 0 [private]`

Expected number of channels of the input image, read from the loaded model.

7.21.4.4 m_expected_image_width

`int64_t OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::m_expected_image_width = 0 [private]`

Width of the CNN used for computation, read from the loaded model.

7.21.4.5 m_number_of_input_elements

`int64_t OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::m_number_of_input_elements = 0 [private]`

Number of input elements of the CNN used for computation, read from the loaded model.

7.21.4.6 m_ort_session

`std::unique_ptr<Ort::Session> OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::m_ort_↩ session [private]`

ONNXRuntime session handle.

7.21.4.7 m_ortenv

`Ort::Env OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::m_ortenv [private]`

ONNXRuntime environment handle.

7.21.4.8 paramPoseEstimatorModel

`const std::string OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::paramPoseEstimatorModel [static], [private]`

Name of the used CNN net, passed from the configuration.

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

- HeadPose3DDFAV2.h

7.22 OFIQ_LIB::modules::measures::HeadSize Class Reference

Implementation of the head size measure.

```
#include <HeadSize.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::HeadSize:

```
┌─────────────────────────────────────────────┐
│  OFIQ_LIB::modules::measures::Measure         │
└─────────────────────────────────────────────┘
                     ▲
                     │
┌─────────────────────────────────────────────┐
│  OFIQ_LIB::modules::measures::HeadSize        │
└─────────────────────────────────────────────┘
```
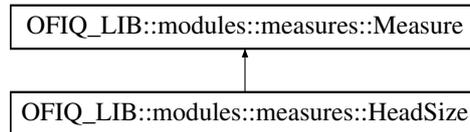
Public Member Functions

- HeadSize (const Configuration &configuration, Session &session)

    Constructor.
- void Execute (OFIQ_LIB::Session &session) override

    Run computation of head size measure,.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

    Constructor.
- virtual ~Measure ()=default

    Destructor.
- virtual std::string GetName () const

    Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const

    Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)

    Inserts the result of a quality assessment in the session object.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

    Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)

    Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)

    Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- const Configuration & configuration

    Reference to the configuration with which the measure constructor has been invoked.

7.22.1   Detailed Description

Implementation of the head size measure.

Head size measure is based on the the distance T between the midpoint between the eyes and the chin and the height of the image. Check ISO/IEC 29794-5 for more information.

7.22.2   Constructor & Destructor Documentation

7.22.2.1   HeadSize()

```
OFIQ_LIB::modules::measures::HeadSize::HeadSize (
            const Configuration & configuration,
            Session & session )
```

Constructor.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.22.3   Member Function Documentation

7.22.3.1   Execute()

```
void OFIQ_LIB::modules::measures::HeadSize::Execute (
            OFIQ_LIB::Session & session )  [override], [virtual]
```

Run computation of head size measure,.

Parameters

session	Session object containing the original facial image and pre-processing results.

Implements OFIQ_LIB::modules::measures::Measure.

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

- HeadSize.h

7.23 OFIQ_LIB::modules::measures::IlluminationUniformity Class Reference

Implementation of the illumination uniformity measure.

```
#include <IlluminationUniformity.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::IlluminationUniformity:
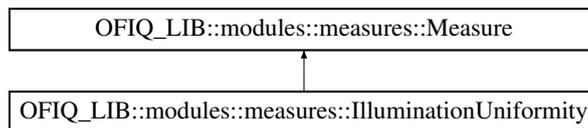
Public Member Functions

- IlluminationUniformity (const Configuration &configuration, Session &session)

    Constructor.
- void Execute (OFIQ_LIB::Session &session) override

    Assesses illumination uniformity.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

    Constructor.
- virtual ∼Measure ()=default

    Destructor.
- virtual std::string GetName () const

    Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const

    Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)

    Inserts the result of a quality assessment in the session object.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

  Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

  Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

  Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)

  Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)

  Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- const Configuration & configuration

  Reference to the configuration with which the measure constructor has been invoked.

7.23.1   Detailed Description

Implementation of the illumination uniformity measure.

Uniformity of the illumination is measured by summing up the minima of the histograms of the left and the right side of the face.

7.23.2   Constructor & Destructor Documentation

7.23.2.1   IlluminationUniformity()

```
OFIQ_LIB::modules::measures::IlluminationUniformity::IlluminationUniformity (
            const Configuration & configuration,
            Session & session )
```

Constructor.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.23.3 Member Function Documentation

7.23.3.1 Execute()

```
void OFIQ_LIB::modules::measures::IlluminationUniformity::Execute (
            OFIQ_LIB::Session & session )  [override], [virtual]
```

Assesses illumination uniformity.

Uniformity of the illumination is measured by summing up the minima of the histograms of the left and the right side of the face.

Parameters

session	Session object computed by the OFIQImpl::performPreprocessing() method.

Implements OFIQ_LIB::modules::measures::Measure.

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

- IlluminationUniformity.h

7.24 OFIQ::Image Struct Reference

Struct representing a single image.

```
#include <ofiq_structs.h>
```

Public Member Functions

- Image ()=default

  Constructor.
- Image (uint16_t width, uint16_t height, uint8_t depth, std::shared_ptr< uint8_t > &data)

  Constructor.
- size_t size () const

  This function returns the size of the image data.

Public Attributes

- uint16_t width { 0 }
- uint16_t height { 0 }
- uint8_t depth { 24 }
- std::shared_ptr< uint8_t > data

7.24.1 Detailed Description

Struct representing a single image.

7.24.2 Constructor & Destructor Documentation

7.24.2.1 Image() [1/2]

```
OFIQ::Image::Image ( ) [default]
```

Constructor.

7.24.2.2 Image() [2/2]

```
OFIQ::Image::Image (
            uint16_t width,
            uint16_t height,
            uint8_t depth,
            std::shared_ptr< uint8_t > & data ) [inline]
```

Constructor.

Parameters

width	of the image.
height	of the image.
depth	of the image
data	of the image.

7.24.3 Member Function Documentation

7.24.3.1 size()

```
size_t OFIQ::Image::size ( ) const [inline]
```

This function returns the size of the image data.

7.24.4 Member Data Documentation

7.24.4.1 data

```
std::shared_ptr<uint8_t> OFIQ::Image::data
```

Managed pointer to raster scanned data. Either RGB color or intensity. If image_depth == 24 this points to 3WH bytes RGBRGBRGB... If image_depth == 8 this points to WH bytes IIIIIII

7.24.4.2 depth

```
uint8_t OFIQ::Image::depth { 24 }
```

Number of bits per pixel. Legal values are 8 and 24.

7.24.4.3 height

```
uint16_t OFIQ::Image::height { 0 }
```

Number of pixels vertically

7.24.4.4 width

```
uint16_t OFIQ::Image::width { 0 }
```

Number of pixels horizontally

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

- ofiq_structs.h

7.25 OFIQ_LIB::modules::measures::InterEyeDistance Class Reference

Implementation of the inter-eye distance measure.

```
#include <InterEyeDistance.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::InterEyeDistance:

```
┌─────────────────────────────────────────────────┐
│     OFIQ_LIB::modules::measures::Measure          │
└─────────────────────────────────────────────────┘
                        ▲
                        │
┌─────────────────────────────────────────────────┐
│  OFIQ_LIB::modules::measures::InterEyeDistance    │
└─────────────────────────────────────────────────┘
```
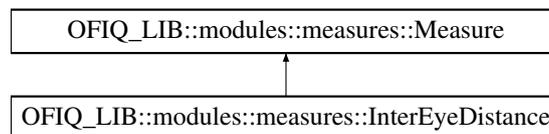
Public Member Functions

- InterEyeDistance (const Configuration &configuration, Session &session)
    Constructor.
- void Execute (OFIQ_LIB::Session &session) override
    Assesses inter-eye distance.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)
    Constructor.
- virtual ∼Measure ()=default
    Destructor.
- virtual std::string GetName () const
    Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const
    Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)
    Inserts the result of a quality assessment in the session object.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

    Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)

    Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)

    Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- const Configuration & configuration

    Reference to the configuration with which the measure constructor has been invoked.

7.25.1 Detailed Description

Implementation of the inter-eye distance measure.

Inter-eye distance assessment is based on computing the Euclidean length of both eyes' centres and multiplication with the secant of the yaw angle computed during pre-processing.

7.25.2 Constructor & Destructor Documentation

7.25.2.1 InterEyeDistance()

```
OFIQ_LIB::modules::measures::InterEyeDistance::InterEyeDistance (
            const Configuration & configuration,
            Session & session )
```

Constructor.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.25.3 Member Function Documentation

7.25.3.1 Execute()

```
void OFIQ_LIB::modules::measures::InterEyeDistance::Execute (
            OFIQ_LIB::Session & session )  [override], [virtual]
```

Assesses inter-eye distance.

Inter-eye distance assessment is based on computing the Euclidean length of both eyes' centres and multiplication with the secant of the yaw angle computed during pre-processing.

Parameters

session	Session object computed by the OFIQImpl::performPreprocessing() method.

Implements OFIQ_LIB::modules::measures::Measure.

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

- InterEyeDistance.h

7.26 OFIQ::Interface Class Reference

The interface to FACE QA implementation.

```
#include <ofiq_lib.h>
```

Inheritance diagram for OFIQ::Interface:

OFIQ::Interface

OFIQ_LIB::OFIQImpl
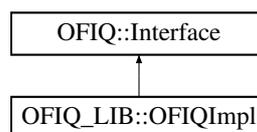
Public Member Functions

- virtual ∼Interface ()=default

    Default Destructor.
- virtual OFIQ::ReturnStatus initialize (const std::string &configDir, const std::string &configFileName)=0

    This function initializes the implementation under test. The implementation under test should set all parameters.
- virtual OFIQ::ReturnStatus scalarQuality (const OFIQ::Image &face, double &quality)=0

    This function takes an image and outputs a quality scalar.
- virtual OFIQ::ReturnStatus vectorQuality (const OFIQ::Image &image, OFIQ::FaceImageQualityAssessment &assessments)=0

    This function takes an image and outputs quality information.

Static Public Member Functions

- static OFIQ_EXPORT std::shared_ptr< Interface > getImplementation ()

    Factory method to return a shared pointer to the Interface object.

7.26.1 Detailed Description

The interface to FACE QA implementation.

Implement this interface by sub-classing this class and implementing each method therein.

7.26.2 Constructor & Destructor Documentation

7.26.2.1 ∼Interface()

```
virtual OFIQ::Interface::∼Interface ( )  [virtual], [default]
```

Default Destructor.

7.26.3 Member Function Documentation

7.26.3.1 getImplementation()

```
static OFIQ_EXPORT std::shared_ptr< Interface > OFIQ::Interface::getImplementation ( )  [static]
```

Factory method to return a shared pointer to the Interface object.

This function is implemented by the submitted library and must return a shared pointer to the Interface object.

This function MUST be implemented.

Note

A possible implementation might be: return (std::make_shared<Implementation>());

Returns

std::shared_ptr<Interface> pointer to the implementation of the interface.

7.26.3.2 initialize()

```
virtual OFIQ::ReturnStatus OFIQ::Interface::initialize (
            const std::string & configDir,
            const std::string & configFileName )  [pure virtual]
```

This function initializes the implementation under test. The implementation under test should set all parameters.

Parameters

in	*configDir*	string representation of the directory containing the configuration file specified by configFileName
in	*configFileName*	An string value encoding the JAXN configuration file name

Returns

OFIQ::ReturnStatus indicating if the initialization was successful.

Implemented in OFIQ_LIB::OFIQImpl.

7.26.3.3 scalarQuality()

```
virtual OFIQ::ReturnStatus OFIQ::Interface::scalarQuality (
        const OFIQ::Image & face,
        double & quality ) [pure virtual]
```

This function takes an image and outputs a quality scalar.

Parameters

in	*face*	Single face image
out	*quality*	A scalar value assessment of image quality. The legal values are [0,100] So, a low value indicates high expected FNMR. A value of -1.0 indicates a failed attempt to calculate a quality score or the value is unassigned.

Returns

OFIQ::ReturnStatus

Implemented in OFIQ_LIB::OFIQImpl.

7.26.3.4 vectorQuality()

```
virtual OFIQ::ReturnStatus OFIQ::Interface::vectorQuality (
        const OFIQ::Image & image,
        OFIQ::FaceImageQualityAssessment & assessments ) [pure virtual]
```

This function takes an image and outputs quality information.

The quality assessment should be performed on the largest detected face.

Parameters

in	*image*	Single face image
out	*assessments*	An ImageQualityAssessments structure. The implementation should populate 1) the bounding box and 2) those items in the QualityAssessments object that the developer chooses to implement 3) face landmarks

Returns

OFIQ::ReturnStatus

Implemented in OFIQ_LIB::OFIQImpl.

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

- ofiq_lib.h

7.27 OFIQ_LIB::modules::landmarks::LandmarkPair Struct Reference

Data container for storing pairs of landmarks.

```
#include <PartExtractor.h>
```

Public Member Functions

- LandmarkPair (OFIQ::LandmarkPoint upper, OFIQ::LandmarkPoint lower)

    Parameterized constructor.

Public Attributes

- OFIQ::LandmarkPoint Upper

    First Landmark.
- OFIQ::LandmarkPoint Lower

    second landmark

7.27.1 Detailed Description

Data container for storing pairs of landmarks.

in some computation special landmarks are bound together via the LandmarkPair struct.

7.27.2 Constructor & Destructor Documentation

7.27.2.1 LandmarkPair()

```
OFIQ_LIB::modules::landmarks::LandmarkPair::LandmarkPair (
            OFIQ::LandmarkPoint upper,
            OFIQ::LandmarkPoint lower )  [inline]
```

Parameterized constructor.

Parameters

in	*upper*	LandmarkPoint of first landmark.
in	*lower*	LandmarkPoint of second landmark.

7.27.3 Member Data Documentation

7.27.3.1 Lower

OFIQ::LandmarkPoint OFIQ_LIB::modules::landmarks::LandmarkPair::Lower

second landmark

7.27.3.2 Upper

OFIQ::LandmarkPoint OFIQ_LIB::modules::landmarks::LandmarkPair::Upper

First Landmark.

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

- PartExtractor.h

7.28 OFIQ::LandmarkPoint Struct Reference

Data structure to describe the x and y coordinate of a landmark.

```
#include <ofiq_structs.h>
```

Public Member Functions

- LandmarkPoint ()=default
    Default constructor.
- LandmarkPoint (int16_t i_x, int16_t i_y)
    Parameterized constructor.

Public Attributes

- int16_t x { -1 }
    x - coordinate
- int16_t y {-1 }
    y - coordinate

7.28.1 Detailed Description

Data structure to describe the x and y coordinate of a landmark.

7.28.2 Constructor & Destructor Documentation

7.28.2.1 LandmarkPoint() [1/2]

```
OFIQ::LandmarkPoint::LandmarkPoint ( )  [default]
```

Default constructor.

7.28.2.2 LandmarkPoint() [2/2]

```
OFIQ::LandmarkPoint::LandmarkPoint (
            int16_t i_x,
            int16_t i_y )  [inline]
```

Parameterized constructor.

Parameters

$i \hookleftarrow$ _$\hookleftarrow$ x	x - coordinate of the landmark.
$i \hookleftarrow$ _$\hookleftarrow$ y	y - coordinate of the landmark.

7.28.3 Member Data Documentation

7.28.3.1 x

```
int16_t OFIQ::LandmarkPoint::x { -1 }
```

x - coordinate

7.28.3.2 y

```
int16_t OFIQ::LandmarkPoint::y {-1 }
```

y - coordinate

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

- ofiq_structs.h

7.29 OFIQ_LIB::modules::measures::Luminance Class Reference

Implementation of two luminance measures.

```
#include <Luminance.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::Luminance:

```
┌─────────────────────────────────────────────┐
│ OFIQ_LIB::modules::measures::Measure          │
└─────────────────────────────────────────────┘
                      ▲
┌─────────────────────────────────────────────┐
│ OFIQ_LIB::modules::measures::Luminance        │
└─────────────────────────────────────────────┘
```
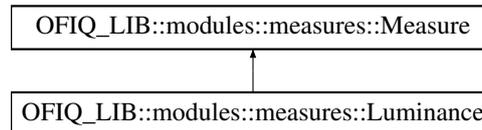
Public Member Functions

- Luminance (const Configuration &configuration, Session &session)

    Constructor.
- void Execute (OFIQ_LIB::Session &session) override

    Assesses luminance mean and luminance variance measures.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

    Constructor.
- virtual ∼Measure ()=default

    Destructor.
- virtual std::string GetName () const

    Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const

    Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)

    Inserts the result of a quality assessment in the session object.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

    Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)

    Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)

    Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- const Configuration & configuration

    Reference to the configuration with which the measure constructor has been invoked.

7.29.1 Detailed Description

Implementation of two luminance measures.

On execution, two measures will be assessed: Luminance mean and luminance variance.

7.29.2 Constructor & Destructor Documentation

7.29.2.1 Luminance()

```
OFIQ_LIB::modules::measures::Luminance::Luminance (
            const Configuration & configuration,
            Session & session )
```

Constructor.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.29.3 Member Function Documentation

7.29.3.1 Execute()

```
void OFIQ_LIB::modules::measures::Luminance::Execute (
            OFIQ_LIB::Session & session )  [override], [virtual]
```

Assesses luminance mean and luminance variance measures.

On execution, two measures will be assessed: Luminance mean and luminance variance.

Parameters

session	Session object computed by the OFIQImpl::performPreprocessing() method.

Implements OFIQ_LIB::modules::measures::Measure.

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

- Luminance.h

7.30 OFIQ_LIB::modules::measures::Measure Class Reference

Base class for measures implemented in OFIQ.

```
#include <Measure.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::Measure:
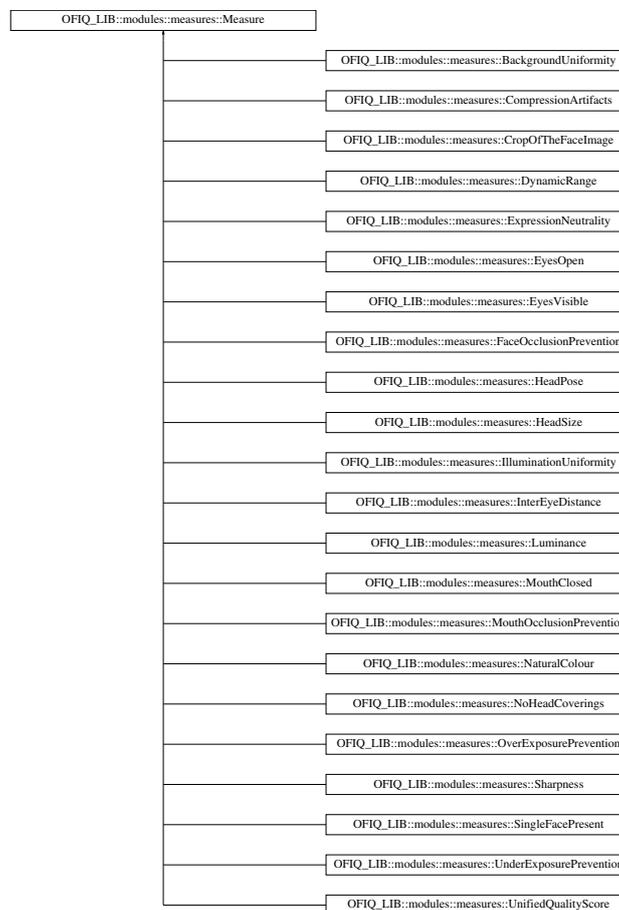
Public Member Functions

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

    Constructor.
- virtual void Execute (OFIQ_LIB::Session &session)=0

    Abstract quality assessment function.
- virtual ∼Measure ()=default

    Destructor.
- virtual std::string GetName () const

Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const
    Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)
    Inserts the result of a quality assessment in the session object.

Protected Member Functions

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)
    Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)
    Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)
    Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)
    Maps a native quality score to a quality component value.

Static Protected Member Functions

- static double Sigmoid (double x, double x0, double w)
    Sigmoid function.

Protected Attributes

- const Configuration & configuration
    Reference to the configuration with which the measure constructor has been invoked.

Static Private Member Functions

- static double ScalarConversion (double rawValue, const SigmoidParameters &par)
    Applies a sigmoid-based quality mapping to a native quality score and outputs the result.
- static std::string GetMeasureName (OFIQ::QualityMeasure measure)
    Returns the name of the specified measure.
- static std::string ExpandKey (std::string_view rawKey)
    Expands the raw key of a measure to the key accessing its Sigmoid-based quality mapping.

Private Attributes

- std::map< std::string, SigmoidParameters, std::less<> > sigmoidMap
    Used to map the measure name to the sigmoid-based quality mapping function.
- OFIQ::QualityMeasure m_measure = OFIQ::QualityMeasure::NotSet
    Value encoding the measure type.

7.30.1 Detailed Description

Base class for measures implemented in OFIQ.

7.30.2 Constructor & Destructor Documentation

7.30.2.1 Measure()

```
OFIQ_LIB::modules::measures::Measure::Measure (
            const Configuration & configuration,
            const OFIQ_LIB::Session & ,
            OFIQ::QualityMeasure measure )  [inline]
```

Constructor.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method
measure	Enum of the measure.

7.30.2.2 ∼Measure()

```
virtual OFIQ_LIB::modules::measures::Measure::~Measure ( )  [virtual], [default]
```

Destructor.

7.30.3 Member Function Documentation

7.30.3.1 AddSigmoid() [1/2]

```
void OFIQ_LIB::modules::measures::Measure::AddSigmoid (
            const std::string & key,
            SigmoidParameters defaultValues )  [protected]
```

Reads sigmoid-function based quality mapping from the configuration.

The parameters are read from the configuration reference member OFIQ_LIB::modules::measures::Measure::configuration. If a parameter is not configured, its default value is chosen from the defaultValues argument.

Parameters

key	Key/name of the measure of which mapping is configured.
defaultValues	Parameters from which default values of non-configured parameters are chosen.

7.30.3.2 AddSigmoid() [2/2]

```
void OFIQ_LIB::modules::measures::Measure::AddSigmoid (
            OFIQ::QualityMeasure measure,
            const SigmoidParameters & defaultValues )  [protected]
```

Reads sigmoid-function based quality mapping from the configuration.

The parameters are read from the configuration reference member OFIQ_LIB::modules::measures::Measure::configuration. If a parameter is not configured, its default value is chosen from the `defaultValues` argument.

Parameters

measure	Enum value encoding the measure for which the mapping is configured.
defaultValues	Parameters from which default values of non-configured parameters are chosen.

7.30.3.3 Execute()

```
virtual void OFIQ_LIB::modules::measures::Measure::Execute (
             OFIQ_LIB::Session & session )  [pure virtual]
```

Abstract quality assessment function.

After quality assessment of the implemented measure, the method should invoke SetQualityMeasure() to insert the result of quality assessment in `session`.

Parameters

session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method.

Implemented in OFIQ_LIB::modules::measures::BackgroundUniformity, OFIQ_LIB::modules::measures::CompressionArtifacts, OFIQ_LIB::modules::measures::CropOfTheFaceImage, OFIQ_LIB::modules::measures::DynamicRange, OFIQ_LIB::modules::measu OFIQ_LIB::modules::measures::EyesOpen, OFIQ_LIB::modules::measures::EyesVisible, OFIQ_LIB::modules::measures::FaceOcclu OFIQ_LIB::modules::measures::HeadPose, OFIQ_LIB::modules::measures::HeadSize, OFIQ_LIB::modules::measures::IlluminationU OFIQ_LIB::modules::measures::InterEyeDistance, OFIQ_LIB::modules::measures::Luminance, OFIQ_LIB::modules::measures::Mou OFIQ_LIB::modules::measures::MouthOcclusionPrevention, OFIQ_LIB::modules::measures::NaturalColour, OFIQ_LIB::modules::measures::NoHeadCoverings, OFIQ_LIB::modules::measures::OverExposurePrevention, OFIQ_LIB::modules::measures::Sharpness, OFIQ_LIB::modules::measures::SingleFacePresent, OFIQ_LIB::modules::measures::Un and OFIQ_LIB::modules::measures::UnifiedQualityScore.

7.30.3.4 ExecuteScalarConversion() [1/2]

```
double OFIQ_LIB::modules::measures::Measure::ExecuteScalarConversion (
             const std::string & key,
             double rawValue )  [protected]
```

Maps a native quality score to a quality component value.

Parameters

key	Key/name of the measure used to read parameters from a private map member.
rawValue	Native quality score.

Returns

Quality component value.

7.30.3.5 ExecuteScalarConversion() [2/2]

```
double OFIQ_LIB::modules::measures::Measure::ExecuteScalarConversion (
            OFIQ::QualityMeasure measure,
            double rawValue )  [protected]
```

Maps a native quality score to a quality component value.

Parameters

measure	Enum value of the measure used to read parameters from a private map member.
rawValue	Native quality score.

Returns

Quality component value.

7.30.3.6 ExpandKey()

```
static std::string OFIQ_LIB::modules::measures::Measure::ExpandKey (
            std::string_view rawKey )  [static], [private]
```

Expands the raw key of a measure to the key accessing its Sigmoid-based quality mapping.

Parameters

rawKey	representation of the measure (e.g., "BackgroundUniformity").

Returns

std::string representation of the key accessing the quality mapping function configuration (e.g., "params.↩
measures.BackgroundUniformity.Sigmoid").

7.30.3.7 GetMeasureName()

```
static std::string OFIQ_LIB::modules::measures::Measure::GetMeasureName (
            OFIQ::QualityMeasure measure )  [static], [private]
```

Returns the name of the specified measure.

Parameters

measure	Enum value of a measure.

Returns

std::string representation of the requested measure.

7.30.3.8 GetName()

```
virtual std::string OFIQ_LIB::modules::measures::Measure::GetName ( ) const  [virtual]
```

Returns the name of the measure.

Unless overwritten, the member m_measure is passed to the private GetMeasureName() method and the result is returned.

Returns

std::string representation of the measures.

7.30.3.9 GetQualityMeasure()

```
virtual OFIQ::QualityMeasure OFIQ_LIB::modules::measures::Measure::GetQualityMeasure ( ) const
[virtual]
```

Returns an enum encoding the measure.

Returns

Enum encoding the measure.

7.30.3.10 ScalarConversion()

```
static double OFIQ_LIB::modules::measures::Measure::ScalarConversion (
            double rawValue,
            const SigmoidParameters & par )  [inline], [static], [private]
```

Applies a sigmoid-based quality mapping to a native quality score and outputs the result.

Before output, the result is checked if it is below 0 or above

1. If it is below 0, then 0 is returned. If it is above 100, then 100 is returned.

Parameters

rawValue	Native quality score.
par	Parameters of the sigmoid-based quality mapping.

Returns

The mapped quality value.

7.30.3.11 SetQualityMeasure()

```
void OFIQ_LIB::modules::measures::Measure::SetQualityMeasure (
            OFIQ_LIB::Session & session,
            OFIQ::QualityMeasure measure,
            double rawValue,
            OFIQ::QualityMeasureReturnCode code )
```

Inserts the result of a quality assessment in the session object.

The method ExecuteScalarConversion() is invoked to map the native quality score to its quality component value.

Parameters

session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method.
measure	Enum value specifying this measure.
rawValue	Native quality score
code	Value indicating whether the quality assessment was computed successfully or otherwise (e.g., failureToAssess).

7.30.3.12 Sigmoid()

```
static double OFIQ_LIB::modules::measures::Measure::Sigmoid (
            double x,
            double x0,
            double w ) [inline], [static], [protected]
```

Sigmoid function.

Parameters

x	Native quality score
x0	Non-zero center point
w	Divisor

Returns

$(1 + \exp((x0 - x)/w))^{-1}$.

7.30.4 Member Data Documentation

7.30.4.1 configuration

```
const Configuration& OFIQ_LIB::modules::measures::Measure::configuration [protected]
```

Reference to the configuration with which the measure constructor has been invoked.

7.30.4.2 m_measure

OFIQ::QualityMeasure OFIQ_LIB::modules::measures::Measure::m_measure = OFIQ::QualityMeasure::NotSet
[private]

Value encoding the measure type.

The value is set to QualityMeasure::NotSet by default which effectively corresponds to a non-specified measure.

7.30.4.3 sigmoidMap

std::map<std::string, SigmoidParameters, std::less<> > OFIQ_LIB::modules::measures::Measure←↩
::sigmoidMap [private]

Used to map the measure name to the sigmoid-based quality mapping function.

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

- Measure.h

7.31 OFIQ_LIB::modules::measures::MeasureFactory Class Reference

Measure factor class.

```
#include <MeasureFactory.h>
```

Public Member Functions

- MeasureFactory ()=delete

Static Public Member Functions

- static std::unique_ptr< Measure > CreateMeasure (const OFIQ::QualityMeasure measure, const Configuration &configuration, OFIQ_LIB::Session &session)
  Requests the creation of a measure implementation.

7.31.1 Detailed Description

Measure factor class.

7.31.2 Constructor & Destructor Documentation

7.31.2.1 MeasureFactory()

OFIQ_LIB::modules::measures::MeasureFactory::MeasureFactory ( ) [delete]

7.31.3 Member Function Documentation

7.31.3.1 CreateMeasure()

```
static std::unique_ptr< Measure > OFIQ_LIB::modules::measures::MeasureFactory::CreateMeasure (
            const OFIQ::QualityMeasure measure,
            const Configuration & configuration,
            OFIQ_LIB::Session & session ) [static]
```

Requests the creation of a measure implementation.

Parameters

measure	Enum value encoding the requested measure.
configuration	Configuration from which measure parameters are read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method.

Attention

The function returns `nullptr` if the request of a measure is not implemented by the function.

If any constructor of a requested measures throws something, the throw is forwarded to this function.

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

- MeasureFactory.h

7.32 OFIQ_LIB::modules::measures::MouthClosed Class Reference

Implementation of the mouth closed measure.

```
#include <MouthClosed.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::MouthClosed:

```
┌─────────────────────────────────────────────┐
│   OFIQ_LIB::modules::measures::Measure        │
└─────────────────────────────────────────────┘
                      ▲
                      │
┌─────────────────────────────────────────────┐
│  OFIQ_LIB::modules::measures::MouthClosed     │
└─────────────────────────────────────────────┘
```
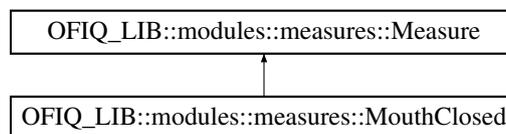
Public Member Functions

- MouthClosed (const Configuration &configuration, Session &session)

  Constructor.
- void Execute (OFIQ_LIB::Session &session) override

  Assesses mouth closeness.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

  Constructor.
- virtual ∼Measure ()=default

  Destructor.
- virtual std::string GetName () const

  Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const

  Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)

  Inserts the result of a quality assessment in the session object.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

    Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)

    Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)

    Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- const Configuration & configuration

    Reference to the configuration with which the measure constructor has been invoked.

7.32.1 Detailed Description

Implementation of the mouth closed measure.

Mouth closed assessment based on computing a ratio from mouth landmarks.

7.32.2 Constructor & Destructor Documentation

7.32.2.1 MouthClosed()

```
OFIQ_LIB::modules::measures::MouthClosed::MouthClosed (
            const Configuration & configuration,
            Session & session )
```

Constructor.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.32.3 Member Function Documentation

7.32.3.1 Execute()

```
void OFIQ_LIB::modules::measures::MouthClosed::Execute (
            OFIQ_LIB::Session & session )  [override], [virtual]
```

Assesses mouth closeness.

Mouth closed assessment based on computing a ratio from mouth landmarks.

Parameters

session	Session object computed by the OFIQImpl::performPreprocessing() method.

See also

> Session::getAlignedFaceLandmarks()

Implements OFIQ_LIB::modules::measures::Measure.

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

- MouthClosed.h

7.33 OFIQ_LIB::modules::measures::MouthOcclusionPrevention Class Reference

Implementation of the mouth occlusion prevention measure.

```
#include <MouthOcclusionPrevention.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::MouthOcclusionPrevention:
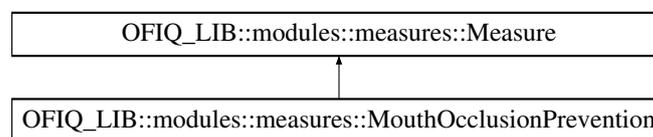
Public Member Functions

- MouthOcclusionPrevention (const Configuration &configuration, Session &session)

    Constructor.

- void Execute (OFIQ_LIB::Session &session) override

    Assesses absence of mouth occlusion.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)
  Constructor.
- virtual ∼Measure ()=default
  Destructor.
- virtual std::string GetName () const
  Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const
  Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)
  Inserts the result of a quality assessment in the session object.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)
  Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)
  Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)
  Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)
  Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)
  Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- const Configuration & configuration
  Reference to the configuration with which the measure constructor has been invoked.

7.33.1 Detailed Description

Implementation of the mouth occlusion prevention measure.

Absence of mouth occlusion is assessed by measuring the coverage of the mouth region from mouth landmarks with the result of face occlusion segmentation computed on pre-processing.

7.33.2 Constructor & Destructor Documentation

7.33.2.1 MouthOcclusionPrevention()

```
OFIQ_LIB::modules::measures::MouthOcclusionPrevention::MouthOcclusionPrevention (
            const Configuration & configuration,
            Session & session )
```

Constructor.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.33.3 Member Function Documentation

7.33.3.1 Execute()

```
void OFIQ_LIB::modules::measures::MouthOcclusionPrevention::Execute (
            OFIQ_LIB::Session & session ) [override], [virtual]
```

Assesses absence of mouth occlusion.

Absence of mouth occlusion is assessed by measuring the coverage of the mouth region from mouth landmarks with the result of face occlusion segmentation computed on pre-processing. Pre-processing results are passed to the method with the `session` parameter.

Parameters

session	Session object computed by the OFIQImpl::performPreprocessing() method.

See also

> FaceOcclusionSegmentation
>
> Session::getAlignedFaceLandmarks()

Implements OFIQ_LIB::modules::measures::Measure.

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

- MouthOcclusionPrevention.h

7.34 OFIQ_LIB::modules::measures::NaturalColour Class Reference

Implementation of the natural colour measure.

```
#include <NaturalColour.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::NaturalColour:

```
┌─────────────────────────────────────────────┐
│   OFIQ_LIB::modules::measures::Measure        │
└─────────────────────────────────────────────┘
                      ▲
                      │
┌─────────────────────────────────────────────┐
│  OFIQ_LIB::modules::measures::NaturalColour   │
└─────────────────────────────────────────────┘
```
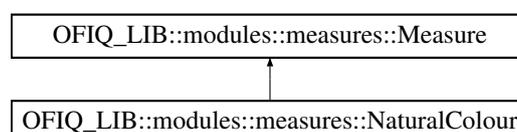
Public Member Functions

- NaturalColour (const Configuration &configuration, Session &session)

    Constructor.
- void Execute (OFIQ_LIB::Session &session) override

    Assesses natural colourness.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

    Constructor.
- virtual ∼Measure ()=default

    Destructor.
- virtual std::string GetName () const

    Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const

    Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)

    Inserts the result of a quality assessment in the session object.

Private Member Functions

- cv::Mat CreateMaskedImage (const OFIQ::FaceLandmarks &landmarks, const cv::Mat &cvImage)

    Creates a mask image from the convex full of the specified landmarks.
- cv::Mat ReduceImageToRegionOfInterest (const cv::Mat maskedImage, const cv::Rect &leftRegionOf↩Interest, const cv::Rect &rightRegionOfInterest)

    Extracts two rectangular regions from an image and returns its concatenation.
- double CalculateScore (double meanChannelA, double meanChannelB)

    Combines two CIELAB values a∗ and b∗ to computed the native quality score.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

    Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)

    Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)

    Sigmoid function.

Protected Attributes inherited from **OFIQ_LIB::modules::measures::Measure**

- const Configuration & configuration

  Reference to the configuration with which the measure constructor has been invoked.

7.34.1 Detailed Description

Implementation of the natural colour measure.

Assessment of the naturalness of the colour based on the conversion of the RGB presentation of the image to the CIELAB colour space.

7.34.2 Constructor & Destructor Documentation

7.34.2.1 NaturalColour()

```
OFIQ_LIB::modules::measures::NaturalColour::NaturalColour (
            const Configuration & configuration,
            Session & session )
```

Constructor.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.34.3 Member Function Documentation

7.34.3.1 CalculateScore()

```
double OFIQ_LIB::modules::measures::NaturalColour::CalculateScore (
            double meanChannelA,
            double meanChannelB ) [private]
```

Combines two CIELAB values a∗ and b∗ to computed the native quality score.

If a^* and b^* are both larger than or equals to 0, then the following formula is applied

$$D = \sqrt{\max(\max(0, 5 - a^*), \max(0, a^* - 25))^2 + \max(\max(0, 5 - b^*), \max(b^* - 35))^2}$$

and D is returned; otherwise, the value 100 is returned.

Parameters

meanChannelA	The CIELAB value a^* input value.
meanChannelB	The CIELAB value b^* input value.

Returns

> Native quality score

7.34.3.2 CreateMaskedImage()

```
cv::Mat OFIQ_LIB::modules::measures::NaturalColour::CreateMaskedImage (
            const OFIQ::FaceLandmarks & landmarks,
            const cv::Mat & cvImage ) [private]
```

Creates a mask image from the convex full of the specified landmarks.

Parameters

landmarks	Facial landmarks.
cvImage	The mask image returned has the same dimension as `cvImage`.

Returns

> Mask image

7.34.3.3 Execute()

```
void OFIQ_LIB::modules::measures::NaturalColour::Execute (
            OFIQ_LIB::Session & session ) [override], [virtual]
```

Assesses natural colourness.

Assessment of the naturalness of the colour based on the conversion of the RGB presentation of the image to the CIELAB colour space.

Parameters

session	Session object computed by the OFIQImpl::performPreprocessing() method.

Implements OFIQ_LIB::modules::measures::Measure.

7.34.3.4 ReduceImageToRegionOfInterest()

```
cv::Mat OFIQ_LIB::modules::measures::NaturalColour::ReduceImageToRegionOfInterest (
            const cv::Mat maskedImage,
            const cv::Rect & leftRegionOfInterest,
            const cv::Rect & rightRegionOfInterest ) [private]
```

Extracts two rectangular regions from an image and returns its concatenation.

Parameters

maskedImage	The input image from which the two regions are extracted.
leftRegionOfInterest	First region
rightRegionOfInterest	Second region

Returns

Concatenation if the requested regions; the first columns correspond to `rightRegionOfInterest` and the last columns correspond to `leftRegionOfInterest`.

Attention

An error occurs if the height of the two requested regions differ.

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

- NaturalColour.h

7.35 OFIQ_LIB::NeuronalNetworkContainer Struct Reference

Neural network container for OFIQ's preprocessing steps.

```
#include <NeuronalNetworkContainer.h>
```

Public Member Functions

- NeuronalNetworkContainer (std::shared_ptr< FaceDetectorInterface > faceDetector, std::shared_ptr< FaceLandmarkExtractorInterface > landmarkExtractor, std::shared_ptr< SegmentationExtractorInterface > segmentationExtractor, std::shared_ptr< PoseEstimatorInterface > poseEstimator, std::shared_ptr< SegmentationExtractorInterface > faceOcclusionExtractor)

    Constructor.

Public Attributes

- std::shared_ptr< FaceDetectorInterface > faceDetector

    Pointer to a FaceDetectorInterface .
- std::shared_ptr< FaceLandmarkExtractorInterface > landmarkExtractor

    Pointer to a FaceLandmarkExtractorInterface .
- std::shared_ptr< SegmentationExtractorInterface > segmentationExtractor

    Pointer to a SegmentationExtractorInterface .
- std::shared_ptr< SegmentationExtractorInterface > faceOcclusionExtractor

    Pointer to a SegmentationExtractorInterface .
- std::shared_ptr< PoseEstimatorInterface > poseEstimator

    Pointer to a SegmentationExtractorInterface .

7.35.1 Detailed Description

Neural network container for OFIQ's preprocessing steps.

7.35.2 Constructor & Destructor Documentation

7.35.2.1 NeuronalNetworkContainer()

```
OFIQ_LIB::NeuronalNetworkContainer::NeuronalNetworkContainer (
            std::shared_ptr< FaceDetectorInterface > faceDetector,
            std::shared_ptr< FaceLandmarkExtractorInterface > landmarkExtractor,
            std::shared_ptr< SegmentationExtractorInterface > segmentationExtractor,
            std::shared_ptr< PoseEstimatorInterface > poseEstimator,
            std::shared_ptr< SegmentationExtractorInterface > faceOcclusionExtractor ) [inline]
```

Constructor.

Parameters

faceDetector	Implementation of a FaceDetectorInterface
landmarkExtractor	Implementation of a FaceLandmarkExtractorInterface
segmentationExtractor	Implementation of a SegmentationExtractorInterface . A pointer to an object instantiated from the FaceParsing class would be valid.
poseEstimator	Implementation of a PoseEstimatorInterface
faceOcclusionExtractor	Implementation of a SegmentationExtractorInterface . A pointer to an object instantiated from the FaceOcclusionSegmentation class would be valid.

7.35.3 Member Data Documentation

7.35.3.1 faceDetector

```
std::shared_ptr<FaceDetectorInterface> OFIQ_LIB::NeuronalNetworkContainer::faceDetector
```

Pointer to a FaceDetectorInterface .

7.35.3.2 faceOcclusionExtractor

```
std::shared_ptr<SegmentationExtractorInterface> OFIQ_LIB::NeuronalNetworkContainer::face←
OcclusionExtractor
```

Pointer to a SegmentationExtractorInterface .

A pointer to an object instantiated from the FaceOcclusionSegmentation class would be valid.

7.35.3.3 landmarkExtractor

```
std::shared_ptr<FaceLandmarkExtractorInterface> OFIQ_LIB::NeuronalNetworkContainer::landmark←
Extractor
```

Pointer to a FaceLandmarkExtractorInterface .

7.35.3.4 poseEstimator

```
std::shared_ptr<PoseEstimatorInterface> OFIQ_LIB::NeuronalNetworkContainer::poseEstimator
```

Pointer to a SegmentationExtractorInterface .

7.35.3.5 segmentationExtractor

```
std::shared_ptr<SegmentationExtractorInterface> OFIQ_LIB::NeuronalNetworkContainer::segmentation←
Extractor
```

Pointer to a SegmentationExtractorInterface .

A pointer to an object instantiated from the FaceParsing class would be valid.

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

- NeuronalNetworkContainer.h

7.36 OFIQ_LIB::modules::measures::NoHeadCoverings Class Reference

Implementation of the no head covering measure.

`#include <NoHeadCoverings.h>`

Inheritance diagram for OFIQ_LIB::modules::measures::NoHeadCoverings:

```
┌─────────────────────────────────────────────────┐
│   OFIQ_LIB::modules::measures::Measure            │
└─────────────────────────────────────────────────┘
                        ▲
┌─────────────────────────────────────────────────┐
│ OFIQ_LIB::modules::measures::NoHeadCoverings      │
└─────────────────────────────────────────────────┘
```
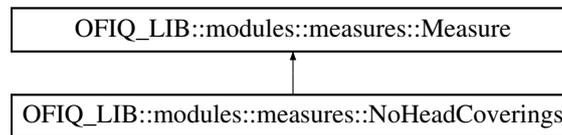
Public Member Functions

- NoHeadCoverings (const Configuration &configuration, Session &session)

    Constructor.
- void Execute (OFIQ_LIB::Session &session) override

    Assesses no head covering.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

    Constructor.
- virtual ∼Measure ()=default

    Destructor.
- virtual std::string GetName () const

    Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const

    Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)

    Inserts the result of a quality assessment in the session object.

Private Attributes

- double threshold

    Threshold.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

    Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)

    Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)

    Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- const Configuration & configuration

    Reference to the configuration with which the measure constructor has been invoked.

7.36.1 Detailed Description

Implementation of the no head covering measure.

The face parsing pre-processing assigns all pixels on the aligned image to one class each encoded by a value between 0 and 19 (inclusively). The values 16 and 18 encode the class *clothing* and *head covering*, respectively. Assessment of no head covering is done on the base of counting all pixels classified as clothing and head covering on the upper part of the aligned facial image and dividing it by the number of all pixels in the aligned image. The ratio is the native quality score. If it exceeds a configurable threshold, a quality the quality component value 0 is used; otherwise, if the ratio is below (or equals) the threshold, a quality of 100 is used.

See also

    FaceParsing

7.36.2 Constructor & Destructor Documentation

7.36.2.1 NoHeadCoverings()

```
OFIQ_LIB::modules::measures::NoHeadCoverings::NoHeadCoverings (
            const Configuration & configuration,
            Session & session )
```

Constructor.

The configuration object can optionally configure the threshold using the `params.measures.NoHead↩Coverings.threshold` key which is 0.02 by default.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.36.3 Member Function Documentation

7.36.3.1 Execute()

```
void OFIQ_LIB::modules::measures::NoHeadCoverings::Execute (
            OFIQ_LIB::Session & session )  [override], [virtual]
```

Assesses no head covering.

The face parsing pre-processing assigns all pixels on the aligned image to one class each encoded by a value between 0 and 19 (inclusively). The values 16 and 18 encode the class *clothing* and *head covering*, respectively. Assessment of no head covering is done on the base of counting all pixels classified as clothing and head covering on the upper part of the aligned facial image and dividing it by the number of all pixels in the aligned image. The ratio is the native quality score. If it exceeds a configurable threshold, a quality the quality component value 0 is used; otherwise, if the ratio is below (or equals) the threshold, a quality of 100 is used.

Parameters

session	Session object computed by the OFIQImpl::performPreprocessing() method.

See also

> FaceParsing

Implements OFIQ_LIB::modules::measures::Measure.

7.36.4 Member Data Documentation

7.36.4.1 threshold

```
double OFIQ_LIB::modules::measures::NoHeadCoverings::threshold  [private]
```

Threshold.

If the native quality score (number of pixels classified as head covering divided by the number of total number of pixels in the aligned image), is below (or equals) this threshold, then a quality of 100 (best) is used; otherwise, a quality of 0 is used.

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

- NoHeadCoverings.h

7.37 OFIQ_LIB::OFIQError Class Reference

Implementation of a custom exception.

```
#include <OFIQError.h>
```

Inheritance diagram for OFIQ_LIB::OFIQError:

```
┌─────────────────────┐
│    std::exception    │
└─────────────────────┘
           ▲
           │
┌─────────────────────┐
│ OFIQ_LIB::OFIQError  │
└─────────────────────┘
```
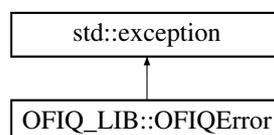
Public Member Functions

- OFIQError (OFIQ::ReturnCode returnCode, std::string_view message)

    Contructor.
- const char ∗ what () const noexcept override

    Getter to the message, overwriting the what method of the base class.
- OFIQ::ReturnCode whatCode () const noexcept

    Getter to the ReturnCode of the QFIQError.

Private Attributes

- OFIQ::ReturnCode returnCode

    Member storing the ReturnCode.
- std::string message

    Member, storing the message passed in the constructor.
- std::string extendedMessage

    The extended message merges the ReturnCode and the message into one string.

7.37.1 Detailed Description

Implementation of a custom exception.

This exception is derived from the standard exception.

7.37.2 Constructor & Destructor Documentation

7.37.2.1 OFIQError()

```
OFIQ_LIB::OFIQError::OFIQError (
            OFIQ::ReturnCode returnCode,
            std::string_view message )
```

Contructor.

Parameters

returnCode	Based on the OFIQ::ReturnCode (see OFIQ::ReturnCode).
message	Message that will be attached to exception.

7.37.3 Member Function Documentation

7.37.3.1 what()

```
const char ∗ OFIQ_LIB::OFIQError::what ( ) const  [inline], [override], [noexcept]
```

Getter to the message, overwriting the what method of the base class.

Returns

const char∗ Pointer to the extended message.

7.37.3.2 whatCode()

[OFIQ::ReturnCode](#) OFIQ_LIB::OFIQError::whatCode ( ) const [inline], [noexcept]

Getter to the ReturnCode of the QFIQError.

Returns

OFIQ::ReturnCode

7.37.4 Member Data Documentation

7.37.4.1 extendedMessage

std::string OFIQ_LIB::OFIQError::extendedMessage [private]

The extended message merges the ReturnCode and the message into one string.

7.37.4.2 message

std::string OFIQ_LIB::OFIQError::message [private]

Member, storing the message passed in the constructor.

7.37.4.3 returnCode

[OFIQ::ReturnCode](#) OFIQ_LIB::OFIQError::returnCode [private]

Member storing the ReturnCode.

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

  • [OFIQError.h](#)

7.38 OFIQ_LIB::OFIQImpl Class Reference

Implementation of the OFIQ_LIB.

#include <ofiq_lib_impl.h>

Inheritance diagram for OFIQ_LIB::OFIQImpl:

```
┌─────────────────────┐
│   OFIQ::Interface   │
└─────────────────────┘
           ▲
           │
┌─────────────────────┐
│ OFIQ_LIB::OFIQImpl  │
└─────────────────────┘
```
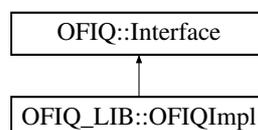
Public Member Functions

- OFIQImpl ()

    Constructor.

- ∼OFIQImpl () override=default

    Destructor.

- OFIQ::ReturnStatus initialize (const std::string &configDir, const std::string &configValue) override

    Initialize the lib by reading the configuration file.

- OFIQ::ReturnStatus scalarQuality (const OFIQ::Image &face, double &quality) override

    Compute an overall quality score for the image provided.

- OFIQ::ReturnStatus vectorQuality (const OFIQ::Image &image, OFIQ::FaceImageQualityAssessment &assessments) override

    Run the computation of all measures set in the configuration.

Public Member Functions inherited from OFIQ::Interface

- virtual ∼Interface ()=default

    Default Destructor.

Private Member Functions

- std::unique_ptr< OFIQ_LIB::modules::measures::Executor > CreateExecutor (Session &session)

    Create a Executor object.

- void CreateNetworks ()

    Create a NeuronalNetworkContainer.

- void performPreprocessing (Session &session)

    Perform the preprocessing.

- void alignFaceImage (Session &session)

    Perform the face alignment.

Private Attributes

- std::unique_ptr< OFIQ_LIB::modules::measures::Executor > m_executorPtr

    Pointer to the executor instance, see OFIQ_LIB::modules::measures::Executor.

- OFIQ::FaceImageQualityAssessment dummyAssement

    required to suit Session constructor

- OFIQ::Image dummyImage

    required to suit Session constructor

- OFIQ_LIB::Session m_emptySession

    required to suit Session constructor

- std::unique_ptr< Configuration > config

    Pointer to the cinfiguration.

- std::unique_ptr< NeuronalNetworkContainer > networks

    Pointer to the different neural network instances, used during the preprocesing.

Additional Inherited Members

Static Public Member Functions inherited from OFIQ::Interface

- static OFIQ_EXPORT std::shared_ptr< Interface > getImplementation ()

    Factory method to return a shared pointer to the Interface object.

7.38.1 Detailed Description

Implementation of the OFIQ_LIB.

7.38.2 Constructor & Destructor Documentation

7.38.2.1 OFIQImpl()

```
OFIQ_LIB::OFIQImpl::OFIQImpl ( )
```

Constructor.

7.38.2.2 ∼OFIQImpl()

```
OFIQ_LIB::OFIQImpl::∼OFIQImpl ( )  [override], [default]
```

Destructor.

7.38.3 Member Function Documentation

7.38.3.1 alignFaceImage()

```
void OFIQ_LIB::OFIQImpl::alignFaceImage (
            Session & session )  [private]
```

Perform the face alignment.

Parameters

session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.38.3.2 CreateExecutor()

```
std::unique_ptr< OFIQ_LIB::modules::measures::Executor > OFIQ_LIB::OFIQImpl::CreateExecutor (
            Session & session )  [private]
```

Create a Executor object.

Parameters

session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

Returns

std::unique_ptr<OFIQ_LIB::modules::measures::Executor>

7.38.3.3 CreateNetworks()

```
void OFIQ_LIB::OFIQImpl::CreateNetworks ( )  [private]
```

Create a NeuronalNetworkContainer.

7.38.3.4 initialize()

```
OFIQ::ReturnStatus OFIQ_LIB::OFIQImpl::initialize (
            const std::string & configDir,
            const std::string & configValue )  [override], [virtual]
```

Initialize the lib by reading the configuration file.

Parameters

configDir	Path to the configuration file.
configValue	Name of the configuration file.

Returns

OFIQ::ReturnStatus

Implements OFIQ::Interface.

7.38.3.5 performPreprocessing()

```
void OFIQ_LIB::OFIQImpl::performPreprocessing (
            Session & session )  [private]
```

Perform the preprocessing.

Parameters

session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.38.3.6 scalarQuality()

```
OFIQ::ReturnStatus OFIQ_LIB::OFIQImpl::scalarQuality (
            const OFIQ::Image & face,
            double & quality )  [override], [virtual]
```

Compute an overall quality score for the image provided.

The overall quality score will be equal to the measure ualityMeasure::UnifiedQualityScore if it is activated. Otherwise, the overall quality score will be the mean of all active measure scores.

Parameters

in	*face*	Input image.
out	*quality*	Computed UnifiedQualityScore.

Returns

OFIQ::ReturnStatus

Implements OFIQ::Interface.

7.38.3.7 vectorQuality()

OFIQ::ReturnStatus OFIQ_LIB::OFIQImpl::vectorQuality (
        const OFIQ::Image & *image,*
        OFIQ::FaceImageQualityAssessment & *assessments*) [override], [virtual]

Run the computation of all measures set in the configuration.

Parameters

in	*image*	Input image.
out	*assessments*	Container to store the resulting scores.

Returns

OFIQ::ReturnStatus

Implements OFIQ::Interface.

7.38.4 Member Data Documentation

7.38.4.1 config

std::unique_ptr<Configuration> OFIQ_LIB::OFIQImpl::config [private]

Pointer to the cinfiguration.

7.38.4.2 dummyAssement

OFIQ::FaceImageQualityAssessment OFIQ_LIB::OFIQImpl::dummyAssement [private]

required to suit Session constructor

7.38.4.3 dummyImage

`OFIQ::Image OFIQ_LIB::OFIQImpl::dummyImage [private]`

required to suit Session constructor

7.38.4.4 m_emptySession

`OFIQ_LIB::Session OFIQ_LIB::OFIQImpl::m_emptySession [private]`

required to suit Session constructor

7.38.4.5 m_executorPtr

`std::unique_ptr<OFIQ_LIB::modules::measures::Executor> OFIQ_LIB::OFIQImpl::m_executorPtr [private]`

Pointer to the executor instance, see OFIQ_LIB::modules::measures::Executor.

7.38.4.6 networks

`std::unique_ptr<NeuronalNetworkContainer> OFIQ_LIB::OFIQImpl::networks [private]`

Pointer to the different neural network instances, used during the preprocesing.

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

- ofiq_lib_impl.h

7.39 ONNXRuntimeSegmentation Class Reference

Helper class to manage the ONNXRuntime session object.

`#include <ONNXRTSegmentation.h>`

Public Member Functions

- ONNXRuntimeSegmentation ()=default

    Constructor.
- ∼ONNXRuntimeSegmentation ()=default

    Destructor.
- void initialize (const std::vector< uint8_t > &i_modelData, int64_t i_imageWidth, int64_t i_imageHeight)

    Public method to generate an ONNXRuntime session object.
- size_t getNumberOfOutputNodes ()

    Get the number of output nodes (results) based on the loaded model.
- std::vector< Ort::Value > run (std::vector< float > &i_netInput)

    Perform the computation.

Private Member Functions

- void init_session (const std::vector< uint8_t > &i_model_data, int64_t i_imageWidth, int64_t i_imageHeight)

  Private method to generate an ONNXRuntime session object.

Private Attributes

- Ort::Env m_ortenv

  Handle to the ONNXRuntime environment variable.
- Ort::MemoryInfo m_memory_info = Ort::MemoryInfo::CreateCpu(OrtDeviceAllocator, OrtMemTypeCPU)

  ONNXRuntime variable to setup the tensors used in ONNXRuntime.
- std::array< int64_t, 4 > m_inputShape

  Description of the shape of the input data expected by the model.
- std::unique_ptr< Ort::Session > m_ort_session

  Handle to the ONNXRuntime session.

7.39.1 Detailed Description

Helper class to manage the ONNXRuntime session object.

Helper class to manage the ONNXRuntime session object. Details can be found on the ONNXRuntime documentation https://onnxruntime.ai/docs/get-started/with-cpp.html.

7.39.2 Constructor & Destructor Documentation

7.39.2.1 ONNXRuntimeSegmentation()

```
ONNXRuntimeSegmentation::ONNXRuntimeSegmentation ( )  [default]
```

Constructor.

7.39.2.2 ∼ONNXRuntimeSegmentation()

```
ONNXRuntimeSegmentation::∼ONNXRuntimeSegmentation ( )  [default]
```

Destructor.

7.39.3 Member Function Documentation

7.39.3.1 getNumberOfOutputNodes()

```
size_t ONNXRuntimeSegmentation::getNumberOfOutputNodes ( )
```

Get the number of output nodes (results) based on the loaded model.

Returns

size_t number of output nodes (results).

7.39.3.2 init_session()

```
void ONNXRuntimeSegmentation::init_session (
            const std::vector< uint8_t > & i_model_data,
            int64_t i_imageWidth,
            int64_t i_imageHeight ) [private]
```

Private method to generate an ONNXRuntime session object.

Parameters

i_model_data	Model data loaded from file.
i_imageWidth	Width of the input image as expected by the model.
i_imageHeight	Height of the input image as expected by the model.

7.39.3.3 initialize()

```
void ONNXRuntimeSegmentation::initialize (
            const std::vector< uint8_t > & i_modelData,
            int64_t i_imageWidth,
            int64_t i_imageHeight )
```

Public method to generate an ONNXRuntime session object.

Parameters

i_modelData	Model data loaded from file.
i_imageWidth	Width of the input image as expected by the model.
i_imageHeight	Height of the input image as expected by the model.

7.39.3.4 run()

```
std::vector< Ort::Value > ONNXRuntimeSegmentation::run (
            std::vector< float > & i_netInput )
```

Perform the computation.

Parameters

i_netInput	Input to the neural net.

Returns

std::vector<Ort::Value> Result of the neural net computation.

7.39.4 Member Data Documentation

7.39.4.1 m_inputShape

```
std::array<int64_t, 4> ONNXRuntimeSegmentation::m_inputShape  [private]
```

Description of the shape of the input data expected by the model.

7.39.4.2 m_memory_info

`Ort::MemoryInfo ONNXRuntimeSegmentation::m_memory_info = Ort::MemoryInfo::CreateCpu(Ort↩`
`DeviceAllocator, OrtMemTypeCPU)  [private]`

ONNXRuntime variable to setup the tensors used in ONNXRuntime.

7.39.4.3 m_ort_session

`std::unique_ptr<Ort::Session> ONNXRuntimeSegmentation::m_ort_session  [private]`

Handle to the ONNXRuntime session.

7.39.4.4 m_ortenv

`Ort::Env ONNXRuntimeSegmentation::m_ortenv  [private]`

Handle to the ONNXRuntime environment variable.

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

- ONNXRTSegmentation.h

7.40 OFIQ_LIB::modules::measures::OverExposurePrevention Class Reference

Implementation of the over-exposure prevention measure.

`#include <OverExposurePrevention.h>`

Inheritance diagram for OFIQ_LIB::modules::measures::OverExposurePrevention:

```
┌─────────────────────────────────────────────────────┐
│      OFIQ_LIB::modules::measures::Measure            │
└─────────────────────────────────────────────────────┘
                         ▲
┌─────────────────────────────────────────────────────┐
│ OFIQ_LIB::modules::measures::OverExposurePrevention  │
└─────────────────────────────────────────────────────┘
```
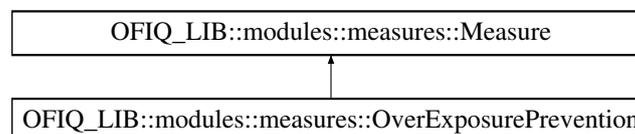
Public Member Functions

- OverExposurePrevention (const Configuration &configuration, Session &session)
  Constructor a new Over Exposure Prevention.
- void Execute (OFIQ_LIB::Session &session) override
  Run the computation of the over-exposure prevention measure.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

  Constructor.
- virtual ∼Measure ()=default

  Destructor.
- virtual std::string GetName () const

  Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const

  Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)

  Inserts the result of a quality assessment in the session object.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

  Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

  Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

  Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)

  Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)

  Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- const Configuration & configuration

  Reference to the configuration with which the measure constructor has been invoked.

7.40.1 Detailed Description

Implementation of the over-exposure prevention measure.

The representation of a face is considered is light if it has a high proportion of pixels that have a high luminance value i.e. hot spots

7.40.2 Constructor & Destructor Documentation

7.40.2.1 OverExposurePrevention()

```
OFIQ_LIB::modules::measures::OverExposurePrevention::OverExposurePrevention (
            const Configuration & configuration,
            Session & session )
```

Constructor a new Over Exposure Prevention.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.40.3 Member Function Documentation

7.40.3.1 Execute()

```
void OFIQ_LIB::modules::measures::OverExposurePrevention::Execute (
            OFIQ_LIB::Session & session )  [override], [virtual]
```

Run the computation of the over-exposure prevention measure.

Parameters

session	Session object computed by the OFIQImpl::performPreprocessing() method.

Implements OFIQ_LIB::modules::measures::Measure.

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

- OverExposurePrevention.h

7.41 OFIQ_LIB::modules::landmarks::PartExtractor Class Reference

Class that provides helper methods for the administration of landmarks.

```
#include <PartExtractor.h>
```

Static Public Member Functions

- static OFIQ::Landmarks getFacePart (const OFIQ::FaceLandmarks &faceLandmarks, FaceParts part)

    Extract the landmarks that correspondend to the requested face part out of a set of provided landmarks.
- static std::vector< LandmarkPair > getPairsForPart (const OFIQ::FaceLandmarks &faceLandmarks, FaceParts part)

    Get LandmarkPairs for a face part.

7.41.1 Detailed Description

Class that provides helper methods for the administration of landmarks.

7.41.2 Member Function Documentation

7.41.2.1 getFacePart()

```
static OFIQ::Landmarks OFIQ_LIB::modules::landmarks::PartExtractor::getFacePart (
            const OFIQ::FaceLandmarks & faceLandmarks,
            FaceParts part )  [static]
```

Extract the landmarks that correspondend to the requested face part out of a set of provided landmarks.

Parameters

in	*faceLandmarks*	Landmarks to be filtered.
	part	Face part of interest.

Returns

OFIQ::Landmarks Filtered landmarks that belong to the requested face part.

7.41.2.2 getPairsForPart()

```
static std::vector< LandmarkPair > OFIQ_LIB::modules::landmarks::PartExtractor::getPairsFor↩
Part (
            const OFIQ::FaceLandmarks & faceLandmarks,
            FaceParts part ) [static]
```

Get LandmarkPairs for a face part.

LandmarkPairs might be used to compute a distance between upper and lower landmark.

Parameters

faceLandmarks	Set of face landmarks.
part	Face part of interest.

Returns

std::vector<LandmarkPair>

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

- PartExtractor.h

7.42 Point2f Struct Reference

Representation of a point with floating point arithmetics.

```
#include <utils.h>
```

Public Attributes

- float x
- float y

7.42.1 Detailed Description

Representation of a point with floating point arithmetics.

7.42.2 Member Data Documentation

7.42.2.1 x

```
float Point2f::x
```

7.42.2.2 y

```
float Point2f::y
```

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

- utils.h

7.43 OFIQ_LIB::Point2i Struct Reference

Representation of a point with integer arithmetics.

```
#include <utils.h>
```

Public Attributes

- int x
- int y

7.43.1 Detailed Description

Representation of a point with integer arithmetics.

7.43.2 Member Data Documentation

7.43.2.1 x

```
int OFIQ_LIB::Point2i::x
```

7.43.2.2 y

```
int OFIQ_LIB::Point2i::y
```

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

- utils.h

7.44 OFIQ_LIB::PoseEstimatorInterface Class Reference

Implementation of the base class for integrating pose estimation algorithms capable of estimating three head orientation angles (yaw, pitch and roll) from a face image.

```
#include <poseEstimators.h>
```

Inheritance diagram for OFIQ_LIB::PoseEstimatorInterface:

```
┌─────────────────────────────────────────────────┐
│          OFIQ_LIB::PoseEstimatorInterface        │
└─────────────────────────────────────────────────┘
                         ▲
┌─────────────────────────────────────────────────────────┐
│  OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2       │
└─────────────────────────────────────────────────────────┘
```
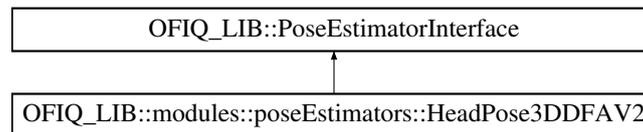
Public Types

- using EulerAngle = std::array<double, 3>

Public Member Functions

- virtual ~PoseEstimatorInterface ()=default

    Standard destructor.
- EulerAngle & estimatePose (OFIQ_LIB::Session &session)

    This function estimates the three head orientation angles.

Protected Member Functions

- virtual void updatePose (OFIQ_LIB::Session &session, EulerAngle &pose)=0

    Call to estimate the head orientations. Has to be implemented in the derived class.

Private Attributes

- std::string lastSessionId

    id of the session that has been used in the latest request, for internal use.
- EulerAngle pose

    Container for storing the estimated head orientations.

7.44.1 Detailed Description

Implementation of the base class for integrating pose estimation algorithms capable of estimating three head orientation angles (yaw, pitch and roll) from a face image.

7.44.2 Member Typedef Documentation

7.44.2.1 EulerAngle

```
using OFIQ_LIB::PoseEstimatorInterface::EulerAngle = std::array<double, 3>
```

7.44.3 Constructor & Destructor Documentation

7.44.3.1 ∼PoseEstimatorInterface()

```
virtual OFIQ_LIB::PoseEstimatorInterface::∼PoseEstimatorInterface ( ) [virtual], [default]
```

Standard destructor.

7.44.4 Member Function Documentation

7.44.4.1 estimatePose()

```
EulerAngle & OFIQ_LIB::PoseEstimatorInterface::estimatePose (
            OFIQ_LIB::Session & session )
```

This function estimates the three head orientation angles.

Parameters

session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.44.4.2 updatePose()

```
virtual void OFIQ_LIB::PoseEstimatorInterface::updatePose (
            OFIQ_LIB::Session & session,
            EulerAngle & pose ) [protected], [pure virtual]
```

Call to estimate the head orientations. Has to be implemented in the derived class.

Parameters

session	Containing the input image for the estimation.
pose	Return the estimated pose.

Implemented in OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2.

7.44.5 Member Data Documentation

7.44.5.1 lastSessionId

```
std::string OFIQ_LIB::PoseEstimatorInterface::lastSessionId [private]
```

id of the session that has been used in the latest request, for internal use.

7.44.5.2 pose

EulerAngle OFIQ_LIB::PoseEstimatorInterface::pose [private]

Container for storing the estimated head orientations.

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

- poseEstimators.h

7.45 OFIQ::QualityMeasureResult Struct Reference

Data structure to handle the results of a quality measure.

```
#include <ofiq_structs.h>
```

Public Member Functions

- QualityMeasureResult ()=default

    Default constructor.
- QualityMeasureResult (double rawScore, double scalar=-1, QualityMeasureReturnCode code=QualityMeasureReturnCode::No

    Parameterized constructor.

Public Attributes

- double rawScore { -1 }

    Raw value as computed by the quality measure implementation.
- double scalar { -1 }

    A scalar value from the interval [0,100] Higher values mean higher quality. A value of -1.0 indicates a failed attempt to calculate a quality score or the value is unassigned.
- QualityMeasureReturnCode code { QualityMeasureReturnCode::NotInitialized }

    Return status code.

7.45.1 Detailed Description

Data structure to handle the results of a quality measure.

7.45.2 Constructor & Destructor Documentation

7.45.2.1 QualityMeasureResult() [1/2]

OFIQ::QualityMeasureResult::QualityMeasureResult ( ) [default]

Default constructor.

7.45.2.2 QualityMeasureResult() [2/2]

```
OFIQ::QualityMeasureResult::QualityMeasureResult (
            double rawScore,
            double scalar = -1,
            QualityMeasureReturnCode code = QualityMeasureReturnCode::NotInitialized ) [inline]
```

Parameterized constructor.

Parameters

in	*rawScore*	Computed raw score.
in	*scalar*	Computed scalar score.
in	*code*	QualityMeasureReturnCode describing the state of the computation.

7.45.3   Member Data Documentation

7.45.3.1   code

QualityMeasureReturnCode OFIQ::QualityMeasureResult::code { QualityMeasureReturnCode::NotInitialized }

Return status code.

7.45.3.2   rawScore

double OFIQ::QualityMeasureResult::rawScore { -1 }

Raw value as computed by the quality measure implementation.

7.45.3.3   scalar

double OFIQ::QualityMeasureResult::scalar { -1 }

A scalar value from the interval [0,100] Higher values mean higher quality. A value of -1.0 indicates a failed attempt to calculate a quality score or the value is unassigned.

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

- ofiq_structs.h

7.46   OFIQ::ReturnStatus Struct Reference

A structure to contain information about a failure by the software under test.

```
#include <ofiq_structs.h>
```

Public Member Functions

- ReturnStatus ()=default
    Default constructor.
- ReturnStatus (const ReturnCode code, const std::string &info="")
    Parameterized constructor.

Public Attributes

- ReturnCode code { ReturnCode::UnknownError }

    Return status code.
- std::string info

    Optional information string.

7.46.1 Detailed Description

A structure to contain information about a failure by the software under test.

An object of this class allows the software to return some information from a function call. The string within this object can be optionally set to provide more information for debugging etc. The status code will be set by the function to Success on success, or one of the other codes on failure.

7.46.2 Constructor & Destructor Documentation

7.46.2.1 ReturnStatus() [1/2]

```
OFIQ::ReturnStatus::ReturnStatus ( )  [default]
```

Default constructor.

7.46.2.2 ReturnStatus() [2/2]

```
OFIQ::ReturnStatus::ReturnStatus (
            const ReturnCode code,
            const std::string & info = "" )  [inline]
```

Parameterized constructor.

Parameters

in	*code*	The return status code; required.
in	*info*	The optional information string.

7.46.3 Member Data Documentation

7.46.3.1 code

```
ReturnCode OFIQ::ReturnStatus::code { ReturnCode::UnknownError }
```

Return status code.

7.46.3.2 info

```
std::string OFIQ::ReturnStatus::info
```

Optional information string.

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

- ofiq_structs.h

7.47 OFIQ_LIB::SegmentationExtractorInterface Class Reference

Base class for the different implementation of segmentation algorithms.

```
#include <segmentations.h>
```

Inheritance diagram for OFIQ_LIB::SegmentationExtractorInterface:

```
                    OFIQ_LIB::SegmentationExtractorInterface

OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation    OFIQ_LIB::modules::segmentations::FaceParsing
```
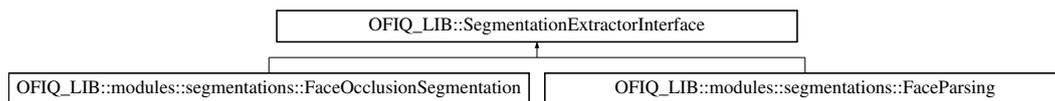
Public Member Functions

- virtual ∼SegmentationExtractorInterface ()=default

    Standard destructor.
- OFIQ::Image & GetMask (OFIQ_LIB::Session &session, modules::segmentations::SegmentClassLabels faceSegment)

    Get a mask of the face region requested.

Protected Member Functions

- virtual OFIQ::Image UpdateMask (OFIQ_LIB::Session &session, modules::segmentations::SegmentClassLabels faceSegment)=0

    Segmentation call that has to be implemented in the derived class.
- std::string GetLastSessionId () const

    Accesses the last session id for this interface.

Private Attributes

- std::string lastSessionId

    id of the session that has been used in the latest request, for internal use.
- std::map< modules::segmentations::SegmentClassLabels, OFIQ::Image > masks

    Container for storing the segmented face region masks.

7.47.1 Detailed Description

Base class for the different implementation of segmentation algorithms.

Base class for the FaceParsing (see OFIQ_LIB::modules::segmentations::FaceParsing) and FaceOcclusion←┘
Segmentation (see OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation)

7.47.2 Constructor & Destructor Documentation

7.47.2.1 ∼SegmentationExtractorInterface()

```
virtual OFIQ_LIB::SegmentationExtractorInterface::∼SegmentationExtractorInterface ( ) [virtual],
[default]
```

Standard destructor.

7.47.3 Member Function Documentation

7.47.3.1 GetLastSessionId()

```
std::string OFIQ_LIB::SegmentationExtractorInterface::GetLastSessionId ( ) const [inline],
[protected]
```

Accesses the last session id for this interface.

Returns

Session id

7.47.3.2 GetMask()

```
OFIQ::Image & OFIQ_LIB::SegmentationExtractorInterface::GetMask (
            OFIQ_LIB::Session & session,
            modules::segmentations::SegmentClassLabels faceSegment )
```

Get a mask of the face region requested.

Parameters

| session | Object containing the relevant data information on the input image. |
| faceSegment | Enum of the face region that is requested. |

Returns

OFIQ::Image& Refernce on the mask of the face region image.

7.47.3.3 UpdateMask()

```
virtual OFIQ::Image OFIQ_LIB::SegmentationExtractorInterface::UpdateMask (
            OFIQ_LIB::Session & session,
            modules::segmentations::SegmentClassLabels faceSegment ) [protected], [pure virtual]
```

Segmentation call that has to be implemented in the derived class.

Parameters

session	Object containing the relevant data information on the input image.
faceSegment	Enum of the face region that is requested

Returns

 OFIQ::Image Segmented face region mask.

Implemented in OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation, and OFIQ_LIB::modules::segmentations::FacePar

7.47.4 Member Data Documentation

7.47.4.1 lastSessionId

```
std::string OFIQ_LIB::SegmentationExtractorInterface::lastSessionId [private]
```

id of the session that has been used in the latest request, for internal use.

7.47.4.2 masks

```
std::map<modules::segmentations::SegmentClassLabels, OFIQ::Image> OFIQ_LIB::Segmentation↩
ExtractorInterface::masks [private]
```

Container for storing the segmented face region masks.

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

 • segmentations.h

7.48 OFIQ_LIB::Session Class Reference

```
#include <Session.h>
```

Public Member Functions

- Session (const OFIQ::Image &image, OFIQ::FaceImageQualityAssessment &assessment)

    Construct a new Session object.
- const OFIQ::Image & image () const

    Acess reference to the input image, connected to this session.
- OFIQ::FaceImageQualityAssessment & assessment ()

    Access reference to the FaceImageQualityAssessment object, connected to this session.
- const std::string & Id () const

    Access to the id connected to this session.
- void setDetectedFaces (const std::vector< OFIQ::BoundingBox > &i_boundingBoxes)

    Set the Detected Faces.
- std::vector< OFIQ::BoundingBox > getDetectedFaces () const

    Get the Detected Faces.
- void setPose (const EulerAngle &i_pose)

    Set the Pose of the input image.
- EulerAngle getPose () const

    Get the Pose of the input image.
- void setLandmarks (const OFIQ::FaceLandmarks &i_landmarks)

    Set the Landmarks detected on the input image.
- OFIQ::FaceLandmarks getLandmarks () const

    Get the Landmarks detected on the input image.
- void setAlignedFaceLandmarks (const OFIQ::FaceLandmarks &i_landmarks)

    Set the Aligned Face Landmarks detected on the aligned image.
- OFIQ::FaceLandmarks getAlignedFaceLandmarks () const

    Get the Aligned Face Landmarks detected on the aligned image.
- void setAlignedFaceTransformationMatrix (const cv::Mat &i_transformationMatrix)

    Set the Aligned Face Transformation Matrix.
- cv::Mat getAlignedFaceTransformationMatrix () const

    Get the Aligned Face Transformation Matrix.
- void setAlignedFace (const cv::Mat &i_alignedFace)

    Set the Aligned Face.
- cv::Mat getAlignedFace () const

    Get the Aligned Face object.
- void setAlignedFaceLandmarkedRegion (const cv::Mat &i_alignedFaceRegion)

    Set the Aligned Face Landmarked Region.
- cv::Mat getAlignedFaceLandmarkedRegion () const

    Get the Aligned Face Landmarked Region.
- void setFaceParsingImage (const cv::Mat &i_parsingImage)

    Set the Face Parsing Image, see OFIQ_LIB::modules::segmentations::FaceParsing).
- cv::Mat getFaceParsingImage () const

    Get the Face Parsing Image, see OFIQ_LIB::modules::segmentations::FaceParsing).
- void setFaceOcclusionSegmentationImage (const cv::Mat &i_segmentationImage)

    Set the Face Occlusion Segmentation Image, see OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation)
- cv::Mat getFaceOcclusionSegmentationImage () const

    Get the Face Occlusion Segmentation Image, see OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation)

Private Member Functions

- std::string GenerateId () const

    Method for generating uuid's for the session.

Private Attributes

- const OFIQ::Image & _image

    Reference to the input image, connected to this session.
- OFIQ::FaceImageQualityAssessment & _assessment

    Refernce to the FaceImageQualityAssessment object, connected to this session.
- std::vector< OFIQ::BoundingBox > detectedFaces

    Container for the faces found on the input image.
- EulerAngle pose

    Container for storing the pose information.
- OFIQ::FaceLandmarks landmarks

    Container for storing the landmark information.
- OFIQ::FaceLandmarks alignedFaceLandmarks

    Container for storing the landmark information of the aligned image.
- cv::Mat alignedFaceTransformationMatrix

    Container for storing the transformation matrix that led to the aligned image.
- cv::Mat alignedFace

    Container for storing the aligned image.
- cv::Mat alignedFacelandmarkedRegion

    Container for storing the landmarks of the aligned face image.
- cv::Mat faceParsingImage

    Container for storing the segmented face image.
- cv::Mat faceOcclusionSegmentationImage

    Container for storing the result of the face occlusion segmented image.
- std::string id

    Container for storing the id of the session.

7.48.1   Detailed Description

@brief The session class is the data container used to distribute the image and additional data,

including the data computed during the pre-processing.

One instance of this class contains the relevant face information used for the computation of the activated measures. Most information is acquired during the pre-processing where the detection of the facial landmarks, the aligned image, etc. is computed.

7.48.2   Constructor & Destructor Documentation

7.48.2.1   Session()

```
OFIQ_LIB::Session::Session (
            const OFIQ::Image & image,
            OFIQ::FaceImageQualityAssessment & assessment )  [inline]
```

Construct a new Session object.

Parameters

image	Input image that shall be analysed.
assessment	Container to staore the computed measures.

7.48.3 Member Function Documentation

7.48.3.1 assessment()

OFIQ::FaceImageQualityAssessment & OFIQ_LIB::Session::assessment ( ) [inline]

Access reference to the FaceImageQualityAssessment object, connected to this session.

Returns

quality assessment object reference.

7.48.3.2 GenerateId()

std::string OFIQ_LIB::Session::GenerateId ( ) const [private]

Method for generating uuid's for the session.

Returns

std::string

7.48.3.3 getAlignedFace()

cv::Mat OFIQ_LIB::Session::getAlignedFace ( ) const

Get the Aligned Face object.

Returns

cv::Mat

7.48.3.4 getAlignedFaceLandmarkedRegion()

cv::Mat OFIQ_LIB::Session::getAlignedFaceLandmarkedRegion ( ) const

Get the Aligned Face Landmarked Region.

Returns

cv::Mat

7.48.3.5 getAlignedFaceLandmarks()

OFIQ::FaceLandmarks OFIQ_LIB::Session::getAlignedFaceLandmarks ( ) const

Get the Aligned Face Landmarks detected on the aligned image.

Returns

OFIQ::FaceLandmarks

7.48.3.6 getAlignedFaceTransformationMatrix()

cv::Mat OFIQ_LIB::Session::getAlignedFaceTransformationMatrix ( ) const

Get the Aligned Face Transformation Matrix.

Returns

cv::Mat

7.48.3.7 getDetectedFaces()

std::vector< OFIQ::BoundingBox > OFIQ_LIB::Session::getDetectedFaces ( ) const

Get the Detected Faces.

Returns

std::vector<OFIQ::BoundingBox> Return the bounding boxes of faces found on the image.

7.48.3.8 getFaceOcclusionSegmentationImage()

cv::Mat OFIQ_LIB::Session::getFaceOcclusionSegmentationImage ( ) const

Get the Face Occlusion Segmentation Image, see OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation)

Returns

cv::Mat

7.48.3.9 getFaceParsingImage()

cv::Mat OFIQ_LIB::Session::getFaceParsingImage ( ) const

Get the Face Parsing Image, see OFIQ_LIB::modules::segmentations::FaceParsing).

Returns

cv::Mat

7.48.3.10 getLandmarks()

`OFIQ::FaceLandmarks OFIQ_LIB::Session::getLandmarks ( ) const`

Get the Landmarks detected on the input image.

Returns

OFIQ::FaceLandmarks

7.48.3.11 getPose()

`EulerAngle OFIQ_LIB::Session::getPose ( ) const`

Get the Pose of the input image.

Returns

EulerAngle Pose of the ipnut image.

7.48.3.12 Id()

`const std::string & OFIQ_LIB::Session::Id ( ) const  [inline]`

Access to the id connected to this session.

Returns

const std::string& Reference to the id of this session.

7.48.3.13 image()

`const OFIQ::Image & OFIQ_LIB::Session::image ( ) const  [inline]`

Acess reference to the input image, connected to this session.

Returns

input image reference.

7.48.3.14 setAlignedFace()

```
void OFIQ_LIB::Session::setAlignedFace (
          const cv::Mat & i_alignedFace )
```

Set the Aligned Face.

Parameters

i_alignedFace	

7.48.3.15 setAlignedFaceLandmarkedRegion()

```
void OFIQ_LIB::Session::setAlignedFaceLandmarkedRegion (
            const cv::Mat & i_alignedFaceRegion )
```

Set the Aligned Face Landmarked Region.

Parameters

i_alignedFaceRegion	

7.48.3.16 setAlignedFaceLandmarks()

```
void OFIQ_LIB::Session::setAlignedFaceLandmarks (
            const OFIQ::FaceLandmarks & i_landmarks )
```

Set the Aligned Face Landmarks detected on the aligned image.

Parameters

i_landmarks	

7.48.3.17 setAlignedFaceTransformationMatrix()

```
void OFIQ_LIB::Session::setAlignedFaceTransformationMatrix (
            const cv::Mat & i_transformationMatrix )
```

Set the Aligned Face Transformation Matrix.

Parameters

i_transformationMatrix	

7.48.3.18 setDetectedFaces()

```
void OFIQ_LIB::Session::setDetectedFaces (
            const std::vector< OFIQ::BoundingBox > & i_boundingBoxes )
```

Set the Detected Faces.

Parameters

i_boundingBoxes	Vector of face bounding boxes found by a face detector.

7.48.3.19 setFaceOcclusionSegmentationImage()

```
void OFIQ_LIB::Session::setFaceOcclusionSegmentationImage (
            const cv::Mat & i_segmentationImage )
```

Set the Face Occlusion Segmentation Image, see OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation)

Parameters

i_segmentationImage	

7.48.3.20 setFaceParsingImage()

```
void OFIQ_LIB::Session::setFaceParsingImage (
            const cv::Mat & i_parsingImage )
```

Set the Face Parsing Image, see OFIQ_LIB::modules::segmentations::FaceParsing).

Parameters

i_parsingImage	

7.48.3.21 setLandmarks()

```
void OFIQ_LIB::Session::setLandmarks (
            const OFIQ::FaceLandmarks & i_landmarks )
```

Set the Landmarks detected on the input image.

Parameters

i_landmarks	

7.48.3.22 setPose()

```
void OFIQ_LIB::Session::setPose (
            const EulerAngle & i_pose )
```

Set the Pose of the input image.

Parameters

i_pose	

7.48.4   Member Data Documentation

7.48.4.1   _assessment

OFIQ::FaceImageQualityAssessment& OFIQ_LIB::Session::_assessment   [private]

Refernce to the FaceImageQualityAssessment object, connected to this session.

7.48.4.2   _image

const OFIQ::Image& OFIQ_LIB::Session::_image   [private]

Reference to the input image, connected to this session.

7.48.4.3   alignedFace

cv::Mat OFIQ_LIB::Session::alignedFace   [private]

Container for storing the aligned image.

7.48.4.4   alignedFacelandmarkedRegion

cv::Mat OFIQ_LIB::Session::alignedFacelandmarkedRegion   [private]

Container for storing the landmarks of the aligned face image.

7.48.4.5   alignedFaceLandmarks

OFIQ::FaceLandmarks OFIQ_LIB::Session::alignedFaceLandmarks   [private]

Container for storing the landmark information of the aligned image.

7.48.4.6   alignedFaceTransformationMatrix

cv::Mat OFIQ_LIB::Session::alignedFaceTransformationMatrix   [private]

Container for storing the transformation matrix that led to the aligned image.

7.48.4.7 detectedFaces

`std::vector<`OFIQ::BoundingBox`> OFIQ_LIB::Session::detectedFaces [private]`

Container for the faces found on the input image.

7.48.4.8 faceOcclusionSegmentationImage

`cv::Mat OFIQ_LIB::Session::faceOcclusionSegmentationImage [private]`

Container for storing the result of the face occlusion segmented image.

7.48.4.9 faceParsingImage

`cv::Mat OFIQ_LIB::Session::faceParsingImage [private]`

Container for storing the segmented face image.

7.48.4.10 id

`std::string OFIQ_LIB::Session::id [private]`

Container for storing the id of the session.

7.48.4.11 landmarks

`OFIQ::FaceLandmarks OFIQ_LIB::Session::landmarks [private]`

Container for storing the landmark information.

7.48.4.12 pose

`EulerAngle OFIQ_LIB::Session::pose [private]`

Container for storing the pose information.

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

- Session.h

7.49 OFIQ_LIB::modules::measures::Sharpness Class Reference

Implemantation of the sharpness measure.

```
#include <Sharpness.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::Sharpness:

```
┌─────────────────────────────────────────────┐
│  OFIQ_LIB::modules::measures::Measure         │
└─────────────────────────────────────────────┘
                      ▲
┌─────────────────────────────────────────────┐
│  OFIQ_LIB::modules::measures::Sharpness       │
└─────────────────────────────────────────────┘
```
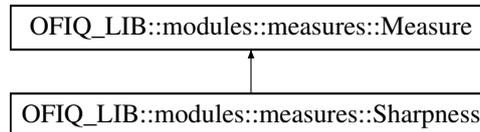
Public Member Functions

- Sharpness (const Configuration &configuration, Session &session)

    Construct a new Sharpness object.
- void Execute (OFIQ_LIB::Session &session) override

    Run computation of the sharpness measure.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

    Constructor.
- virtual ∼Measure ()=default

    Destructor.
- virtual std::string GetName () const

    Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const

    Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)

    Inserts the result of a quality assessment in the session object.

Private Member Functions

- void GetCroppedImages (Session &session, cv::Mat &faceCrop, cv::Mat &maskCrop, bool useAligned, float faceRegionAlpha)

    Get the cropped face region.
- cv::Mat GetClassifierFocusFeatures (cv::Mat &image, cv::Mat &mask, bool applyBlur)

    Computation of the input features using different edge detectors.

Private Attributes

- std::string modelFile

    Name of the random forest model, extracted from the configuration file.
- std::shared_ptr< cv::ml::RTrees > rtree

    Instance of the random forest model.
- bool useAligned

    The sharpness measure can be computed on the aligned or the original image. useAligned set to true will run the computation on the aligned image. The member is read from the configuration file.
- double faceRegionAlpha

    For faceRegionAlpha = 0, the algorithm uses the inner face region. For faceRegionAlpha = 0.85, the algorithm uses the extended face region as specified for the FaceOcclusionin FRVT Quality.
- int numTrees

    This member stores the number of trees used for the random forest. Internal use only.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

    Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)

    Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)

    Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- const Configuration & configuration

    Reference to the configuration with which the measure constructor has been invoked.

7.49.1 Detailed Description

Implemantation of the sharpness measure.

This quality component can be used to efficiently choose the better focused face portrait among several face samples of the same biometric capture subject. It should not be used to perform an absolute sharpness assessment if only one sample is available

7.49.2 Constructor & Destructor Documentation

7.49.2.1 Sharpness()

```
OFIQ_LIB::modules::measures::Sharpness::Sharpness (
          const Configuration & configuration,
          Session & session )
```

Construct a new Sharpness object.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.49.3 Member Function Documentation

7.49.3.1 Execute()

```
void OFIQ_LIB::modules::measures::Sharpness::Execute (
            OFIQ_LIB::Session & session ) [override], [virtual]
```

Run computation of the sharpness measure.

Parameters

session	Session object computed by the OFIQImpl::performPreprocessing() method.

Implements OFIQ_LIB::modules::measures::Measure.

7.49.3.2 GetClassifierFocusFeatures()

```
cv::Mat OFIQ_LIB::modules::measures::Sharpness::GetClassifierFocusFeatures (
            cv::Mat & image,
            cv::Mat & mask,
            bool applyBlur ) [private]
```

Computation of the input features using different edge detectors.

Parameters

image	Input image.
mask	Input region of the face.
applyBlur	Wheter or not rub a GaussianBlur before the edge detection.

Returns

cv::Mat Container storing the results of the different edge detectors.

7.49.3.3 GetCroppedImages()

```
void OFIQ_LIB::modules::measures::Sharpness::GetCroppedImages (
            Session & session,
            cv::Mat & faceCrop,
            cv::Mat & maskCrop,
```

```
                    bool useAligned,
                    float faceRegionAlpha ) [private]
```

Get the cropped face region.

Parameters

session	Data container.
faceCrop	Computed crop of the face.
maskCrop	Mask used for the cropping. Will be computed in the method.
useAligned	Switch for using the aligned image.
faceRegionAlpha	Enlarge the face region by passing this parameter.

7.49.4 Member Data Documentation

7.49.4.1 faceRegionAlpha

```
double OFIQ_LIB::modules::measures::Sharpness::faceRegionAlpha [private]
```

For faceRegionAlpha = 0, the algorithm uses the inner face region. For faceRegionAlpha = 0.85, the algorithm uses the extended face region as specified for the FaceOcclusionin FRVT Quality.

7.49.4.2 modelFile

```
std::string OFIQ_LIB::modules::measures::Sharpness::modelFile [private]
```

Name of the random forest model, extracted from the configuration file.

7.49.4.3 numTrees

```
int OFIQ_LIB::modules::measures::Sharpness::numTrees [private]
```

This member stores the number of trees used for the random forest. Internal use only.

7.49.4.4 rtree

```
std::shared_ptr<cv::ml::RTrees> OFIQ_LIB::modules::measures::Sharpness::rtree [private]
```

Instance of the random forest model.

7.49.4.5 useAligned

```
bool OFIQ_LIB::modules::measures::Sharpness::useAligned [private]
```

The sharpness measure can be computed on the aligned or the original image. useAligned set to true will run the computation on the aligned image. The member is read from the configuration file.

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

- Sharpness.h

7.50 OFIQ_LIB::modules::measures::SigmoidParameters Struct Reference

Parameters of the sigmoid function based quality mapping.

```
#include <Measure.h>
```

Public Member Functions

- SigmoidParameters ()

  Default constructor.
- void setInverse ()

  Sets this quality mapping to a smaller-is-better variant.
- void Reset ()

  Resets the members of the quality mapping to their default values.

Public Attributes

- double h

  Scale factor.
- double a

  Constant shift.
- double s

  Signed weight for sigmoid part.
- double x0

  Center point in sigmoid part.
- double w

  Divisor in sigmoid part.
- bool round

  Flag controlling if the compiler's native rounding function (`std::round`) is applied.

7.50.1 Detailed Description

Parameters of the sigmoid function based quality mapping.

A sigmoid-based quality mapping is the following function

$$Q(x) = h \cdot (a + s \cdot \text{sigmoid}(x, x_0, w))$$

where

$$\text{sigmoid}(x, x_0, w) = (1 + \exp((x_0 - x)/w)^{-1}.$$

Q can be used to map a native quality score x to a value between 0 and 100. The other symbols denote parameters that can be configured using the struct.

7.50.2 Constructor & Destructor Documentation

7.50.2.1 SigmoidParameters()

```
OFIQ_LIB::modules::measures::SigmoidParameters::SigmoidParameters ( )  [inline]
```

Default constructor.

After construction, all members are set to their default values.

7.50.3 Member Function Documentation

7.50.3.1 Reset()

```
void OFIQ_LIB::modules::measures::SigmoidParameters::Reset ( )  [inline]
```

Resets the members of the quality mapping to their default values.

7.50.3.2 setInverse()

```
void OFIQ_LIB::modules::measures::SigmoidParameters::setInverse ( )  [inline]
```

Sets this quality mapping to a smaller-is-better variant.

If the parameters a is 0 and s is, then this quality mapping is in larger-is-better-semantics. For those mappings, the method can be used to set the mapping to its smaller-is- better counterpart by setting a to 1 and s to -1. This is used by some measures to conveniently set a quality mapping.

7.50.4 Member Data Documentation

7.50.4.1 a

```
double OFIQ_LIB::modules::measures::SigmoidParameters::a
```

Constant shift.

The default value is 0.

7.50.4.2 h

```
double OFIQ_LIB::modules::measures::SigmoidParameters::h
```

Scale factor.

The default value is 100.

7.50.4.3 round

```
bool OFIQ_LIB::modules::measures::SigmoidParameters::round
```

Flag controlling if the compiler's native rounding function (`std::round`) is applied.

The default value is `true`.

7.50.4.4 s

```
double OFIQ_LIB::modules::measures::SigmoidParameters::s
```

Signed weight for sigmoid part.

The default value is 1.

7.50.4.5 w

```
double OFIQ_LIB::modules::measures::SigmoidParameters::w
```

Divisor in sigmoid part.

The default value of 0.7 has been chosen arbitrarily and should specified when a mapping is configured.

7.50.4.6 x0

```
double OFIQ_LIB::modules::measures::SigmoidParameters::x0
```

Center point in sigmoid part.

The default value of 4 has been chosen arbitrarily and should specified when a mapping is configured.

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

- Measure.h

7.51 OFIQ_LIB::modules::measures::SingleFacePresent Class Reference

Implementation of the single face present measure.

```
#include <SingleFacePresent.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::SingleFacePresent:

OFIQ_LIB::modules::measures::Measure

OFIQ_LIB::modules::measures::SingleFacePresent
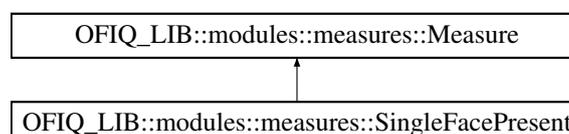
Public Member Functions

- SingleFacePresent (const Configuration &configuration, Session &session)

    Construct a new Single Face Present object.
- void Execute (OFIQ_LIB::Session &session) override

    Run computation of the single face present analysis.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

    Constructor.
- virtual ∼Measure ()=default

    Destructor.
- virtual std::string GetName () const

    Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const

    Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)

    Inserts the result of a quality assessment in the session object.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

    Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)

    Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)

    Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- const Configuration & configuration

    Reference to the configuration with which the measure constructor has been invoked.

7.51.1 Detailed Description

Implementation of the single face present measure.

Single face present measure shall ensure that only one face is visible on the image.

7.51.2 Constructor & Destructor Documentation

7.51.2.1 SingleFacePresent()

```
OFIQ_LIB::modules::measures::SingleFacePresent::SingleFacePresent (
            const Configuration & configuration,
            Session & session )
```

Construct a new Single Face Present object.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.51.3 Member Function Documentation

7.51.3.1 Execute()

```
void OFIQ_LIB::modules::measures::SingleFacePresent::Execute (
            OFIQ_LIB::Session & session ) [override], [virtual]
```

Run computation of the single face present analysis.

Parameters

session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

Implements OFIQ_LIB::modules::measures::Measure.

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

- SingleFacePresent.h

7.52 OFIQ_LIB::modules::detectors::SSDFaceDetector Class Reference

Implementation of a face detector using the SSD face detector CNN.

```
#include <opencv_ssd_face_detector.h>
```

Inheritance diagram for OFIQ_LIB::modules::detectors::SSDFaceDetector:

```
┌─────────────────────────────────────────────┐
│        OFIQ_LIB::FaceDetectorInterface        │
└─────────────────────────────────────────────┘
                      ▲
                      │
┌─────────────────────────────────────────────┐
│  OFIQ_LIB::modules::detectors::SSDFaceDetector │
└─────────────────────────────────────────────┘
```
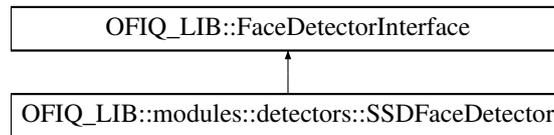
Public Member Functions

- SSDFaceDetector (const Configuration &config)

    Constructor a new SSDFaceDetector.
- ∼SSDFaceDetector () override=default

    Destructor of the SSDFaceDetector.

Public Member Functions inherited from OFIQ_LIB::FaceDetectorInterface

- virtual ∼FaceDetectorInterface ()=default

    Destroy the Face Detector Interface object.
- std::vector< OFIQ::BoundingBox > detectFaces (OFIQ_LIB::Session &session)

    This function detects faces in given image.

Protected Member Functions

- std::vector< OFIQ::BoundingBox > UpdateFaces (OFIQ_LIB::Session &session) override

    Implementation of the face detection method.

Private Attributes

- std::shared_ptr< cv::dnn::Net > dnnNet {nullptr}

    Instance of an opencv dnn::Net.
- double confidenceThreshold

    Confidence threshold used for the face detection. The value is read from the configuration file.
- double padding

    Add padding around the image (faceImage.width ∗ padding; faceImage.height ∗ padding;)
- double minimalRelativeFaceSize

    Filter threshold for removing to small face found on the image. This value is read from the configuration file.

7.52.1   Detailed Description

Implementation of a face detector using the SSD face detector CNN.

7.52.2   Constructor & Destructor Documentation

7.52.2.1   SSDFaceDetector()

```
OFIQ_LIB::modules::detectors::SSDFaceDetector::SSDFaceDetector (
            const Configuration & config )  [explicit]
```

Constructor a new SSDFaceDetector.

Parameters

config	

7.52.2.2 ∼SSDFaceDetector()

```
OFIQ_LIB::modules::detectors::SSDFaceDetector::∼SSDFaceDetector ( ) [override], [default]
```

Destructor of the SSDFaceDetector.

7.52.3 Member Function Documentation

7.52.3.1 UpdateFaces()

```
std::vector< OFIQ::BoundingBox > OFIQ_LIB::modules::detectors::SSDFaceDetector::UpdateFaces (
            OFIQ_LIB::Session & session ) [override], [protected], [virtual]
```

Implementation of the face detection method.

Parameters

session	Session object computed by the OFIQImpl::performPreprocessing() method.

Returns

std::vector<OFIQ::BoundingBox> Bounding boxes of the detected faces

Implements OFIQ_LIB::FaceDetectorInterface.

7.52.4 Member Data Documentation

7.52.4.1 confidenceThreshold

```
double OFIQ_LIB::modules::detectors::SSDFaceDetector::confidenceThreshold [private]
```

Confidence threshold used for the face detection. The value is read from the configuration file.

7.52.4.2 dnnNet

```
std::shared_ptr<cv::dnn::Net> OFIQ_LIB::modules::detectors::SSDFaceDetector::dnnNet {nullptr}
[private]
```

Instance of an opencv dnn::Net.

7.52.4.3 minimalRelativeFaceSize

`double OFIQ_LIB::modules::detectors::SSDFaceDetector::minimalRelativeFaceSize [private]`

Filter threshold for removing to small face found on the image. This value is read from the configuration file.

7.52.4.4 padding

`double OFIQ_LIB::modules::detectors::SSDFaceDetector::padding [private]`

Add padding around the image (faceImage.width ∗ padding; faceImage.height ∗ padding;)

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

- opencv_ssd_face_detector.h

7.53 OFIQ_LIB::modules::measures::UnderExposurePrevention Class Reference

Implementation of the under-exposure prevention measure.

`#include <UnderExposurePrevention.h>`

Inheritance diagram for OFIQ_LIB::modules::measures::UnderExposurePrevention:

```
┌────────────────────────────────────────────────────┐
│      OFIQ_LIB::modules::measures::Measure            │
└────────────────────────────────────────────────────┘
                         ▲
                         │
┌────────────────────────────────────────────────────┐
│ OFIQ_LIB::modules::measures::UnderExposurePrevention │
└────────────────────────────────────────────────────┘
```
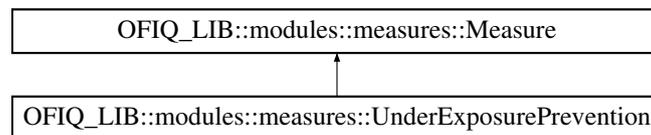
Public Member Functions

- UnderExposurePrevention (const Configuration &configuration, Session &session)

    Constructor a new Under Exposure Prevention object.
- void Execute (OFIQ_LIB::Session &session) override

    Run the computation of the under-exposure prevention measure.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

    Constructor.
- virtual ∼Measure ()=default

    Destructor.
- virtual std::string GetName () const

    Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const

    Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)

    Inserts the result of a quality assessment in the session object.

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- • void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

  Reads sigmoid-function based quality mapping from the configuration.
- • void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

  Reads sigmoid-function based quality mapping from the configuration.
- • double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

  Maps a native quality score to a quality component value.
- • double ExecuteScalarConversion (const std::string &key, double rawValue)

  Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- • static double Sigmoid (double x, double x0, double w)

  Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- • const Configuration & configuration

  Reference to the configuration with which the measure constructor has been invoked.

7.53.1 Detailed Description

Implementation of the under-exposure prevention measure.

The representation of a face is considered too dark if it has a high proportion of pixels that have a low luminance value.

7.53.2 Constructor & Destructor Documentation

7.53.2.1 UnderExposurePrevention()

```
OFIQ_LIB::modules::measures::UnderExposurePrevention::UnderExposurePrevention (
            const Configuration & configuration,
            Session & session )
```

Constructor a new Under Exposure Prevention object.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.53.3 Member Function Documentation

7.53.3.1 Execute()

```
void OFIQ_LIB::modules::measures::UnderExposurePrevention::Execute (
            OFIQ_LIB::Session & session )  [override], [virtual]
```

Run the computation of the under-exposure prevention measure.

Parameters

session	Session object computed by the OFIQImpl::performPreprocessing() method.

Implements OFIQ_LIB::modules::measures::Measure.

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

- UnderExposurePrevention.h

7.54 OFIQ_LIB::modules::measures::UnifiedQualityScore Class Reference

Implementation of the unified quality measure.

```
#include <UnifiedQualityScore.h>
```

Inheritance diagram for OFIQ_LIB::modules::measures::UnifiedQualityScore:
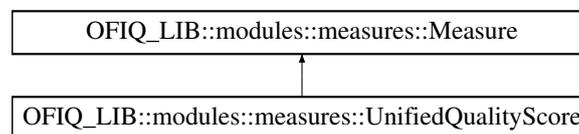
Public Member Functions

- UnifiedQualityScore (const Configuration &configuration, Session &session)

    Construct a new Unified Quality Score object.
- void Execute (OFIQ_LIB::Session &session) override

    Run the computation on the measure.

Public Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- Measure (const Configuration &configuration, const OFIQ_LIB::Session &, OFIQ::QualityMeasure measure)

    Constructor.
- virtual ∼Measure ()=default

    Destructor.
- virtual std::string GetName () const

    Returns the name of the measure.
- virtual OFIQ::QualityMeasure GetQualityMeasure () const

    Returns an enum encoding the measure.
- void SetQualityMeasure (OFIQ_LIB::Session &session, OFIQ::QualityMeasure measure, double rawValue, OFIQ::QualityMeasureReturnCode code)

    Inserts the result of a quality assessment in the session object.

Private Attributes

- ONNXRuntimeSegmentation m_onnxRuntimeEnv

    Instance of the neural network (iResNet50 model M).

Additional Inherited Members

Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- void AddSigmoid (OFIQ::QualityMeasure measure, const SigmoidParameters &defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- void AddSigmoid (const std::string &key, SigmoidParameters defaultValues)

    Reads sigmoid-function based quality mapping from the configuration.
- double ExecuteScalarConversion (OFIQ::QualityMeasure measure, double rawValue)

    Maps a native quality score to a quality component value.
- double ExecuteScalarConversion (const std::string &key, double rawValue)

    Maps a native quality score to a quality component value.

Static Protected Member Functions inherited from OFIQ_LIB::modules::measures::Measure

- static double Sigmoid (double x, double x0, double w)

    Sigmoid function.

Protected Attributes inherited from OFIQ_LIB::modules::measures::Measure

- const Configuration & configuration

    Reference to the configuration with which the measure constructor has been invoked.

7.54.1 Detailed Description

Implementation of the unified quality measure.

The quality score refers to the requirements in clause 5.4.8 in ISO/IEC 19794-5:2011 frontal image type (relevant for EU-EES implementing decision 2019/329) and in clause 7.7 in ISO/IEC 39794-1 (relevant for UC1 specified in ICAO Document 9303)

7.54.2 Constructor & Destructor Documentation

7.54.2.1 UnifiedQualityScore()

```
OFIQ_LIB::modules::measures::UnifiedQualityScore::UnifiedQualityScore (
            const Configuration & configuration,
            Session & session )
```

Construct a new Unified Quality Score object.

Parameters

configuration	Configuration object from which measure-related configuration is read.
session	Session object containing the original facial image and pre-processing results computed by the OFIQImpl::performPreprocessing() method

7.54.3 Member Function Documentation

7.54.3.1 Execute()

```
void OFIQ_LIB::modules::measures::UnifiedQualityScore::Execute (
            OFIQ_LIB::Session & session )  [override], [virtual]
```

Run the computation on the measure.

The algorithm uses a iResNet50 model M from https://github.com/IrvingMeng/MagFace trained on MS1MV2 with MagFace loss without DDP parallelisation. The algorithm takes as input the image I output by the alignment algorithm.

Parameters

session	Session object computed by the OFIQImpl::performPreprocessing() method.

Implements OFIQ_LIB::modules::measures::Measure.

7.54.4 Member Data Documentation

7.54.4.1 m_onnxRuntimeEnv

```
ONNXRuntimeSegmentation OFIQ_LIB::modules::measures::UnifiedQualityScore::m_onnxRuntimeEnv
[private]
```

Instance of the neural network (iResNet50 model M).

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

- UnifiedQualityScore.h

Chapter 8

File Documentation

8.1 mainpage.h File Reference

This header file is for generating the doxygen documentation for OFIQ.

8.1.1 Detailed Description

This header file is for generating the doxygen documentation for OFIQ.

Copyright

 Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

 OFIQ development team

8.2 mainpage.h

Go to the documentation of this file.
```
00001
01147 #pragma once
```

8.3 ofiq_lib.h File Reference

Class describing the interface to the OFIQ.

```
#include <cstdint>
#include <string>
#include <vector>
#include <ofiq_structs.h>
```

Classes

- class OFIQ::Interface

    The interface to FACE QA implementation.

Namespaces

- namespace OFIQ

    Namespace for OFIQ API.

Macros

- #define OFIQ_EXPORT

8.3.1 Detailed Description

Class describing the interface to the OFIQ.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.3.2 Macro Definition Documentation

8.3.2.1 OFIQ_EXPORT

```
#define OFIQ_EXPORT
```

8.4 ofiq_lib.h

Go to the documentation of this file.
```
00001
00027 #ifndef OFIQ_LIB_H
00028 #define OFIQ_LIB_H
00029
00030 #include <cstdint>
00031 #include <string>
00032 #include <vector>
00033
00034 #include <ofiq_structs.h>
00035
00036 #ifdef _WIN32
00037 #    ifdef OFIQ_EXPORTS
00038 #        define OFIQ_EXPORT __declspec(dllexport)
00039 #    else
00040 #        define OFIQ_EXPORT __declspec(dllimport)
00041 #    endif
00042 #else
00043 #    define OFIQ_EXPORT
00044 #endif
00045
00049 namespace OFIQ
00050 {
00051
00059     class Interface
00060     {
00061     public:
00066         virtual ~Interface() = default;
00067
00080         virtual OFIQ::ReturnStatus
00081             initialize(const std::string& configDir, const std::string& configFileName) = 0;
00082
00095         virtual OFIQ::ReturnStatus scalarQuality(const OFIQ::Image& face, double& quality) = 0;
00096
00115         virtual OFIQ::ReturnStatus vectorQuality(
00116             const OFIQ::Image& image, OFIQ::FaceImageQualityAssessment& assessments) = 0;
00117
00133         OFIQ_EXPORT static std::shared_ptr<Interface> getImplementation();
00134     };
00135 }
00136
00137 #endif /* OFIQ_LIB_H */
```

8.5 ofiq_lib_impl.h File Reference

Implementation of the OFIQ_LIB.

```
#include "Configuration.h"
#include "Executor.h"
#include "ofiq_lib.h"
#include "NeuronalNetworkContainer.h"
```

Classes

- class OFIQ_LIB::OFIQImpl

    Implementation of the OFIQ_LIB.

Namespaces

- namespace OFIQ_LIB

  Namespace for OFIQ implementations.

8.5.1 Detailed Description

Implementation of the OFIQ_LIB.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.6 ofiq_lib_impl.h

Go to the documentation of this file.
```
00001
00027 #ifndef OFIQ_LIB_IMPL_H
00028 #define OFIQ_LIB_IMPL_H
00029
00030 #include "Configuration.h"
00031 #include "Executor.h"
00032 #include "ofiq_lib.h"
00033 #include "NeuronalNetworkContainer.h"
00034
00038 namespace OFIQ_LIB
00039 {
00044     class OFIQImpl : public OFIQ::Interface
00045     {
00046     public:
00051         OFIQImpl();
00052
00057         ~OFIQImpl() override = default;
00058
00066         OFIQ::ReturnStatus
00067             initialize(const std::string& configDir, const std::string& configValue) override;
00068
00069
00078         OFIQ::ReturnStatus scalarQuality(const OFIQ::Image& face, double& quality) override;
00079
00080
00088         OFIQ::ReturnStatus vectorQuality(
00089             const OFIQ::Image& image, OFIQ::FaceImageQualityAssessment& assessments) override;
```

```
00090
00091     private:
00096         std::unique_ptr<OFIQ_LIB::modules::measures::Executor> m_executorPtr;
00097
00102         OFIQ::FaceImageQualityAssessment dummyAssement;
00103
00108         OFIQ::Image dummyImage;
00109
00114         OFIQ_LIB::Session m_emptySession;
00115
00116
00121         std::unique_ptr<Configuration> config;
00122
00127         std::unique_ptr<NeuronalNetworkContainer> networks;
00128
00137         std::unique_ptr<OFIQ_LIB::modules::measures::Executor> CreateExecutor(Session& session);
00138
00139
00144         void CreateNetworks();
00145
00153         void performPreprocessing(Session& session);
00154
00162         void alignFaceImage(Session& session);
00163     };
00164 }
00165
00166 #endif /* OFIQ_LIB_IMPL_H */
```

8.7 ofiq_structs.h File Reference

PRovides several helper classes, enums and functions used in the OFIQ framework.

```
#include <cstdint>
#include <iostream>
#include <map>
#include <memory>
#include <string>
#include <utility>
#include <vector>
```

Classes

- struct OFIQ::Image

  Struct representing a single image.
- struct OFIQ::ReturnStatus

  A structure to contain information about a failure by the software under test.
- struct OFIQ::QualityMeasureResult

  Data structure to handle the results of a quality measure.
- struct OFIQ::BoundingBox

  Data structure for desciding bounding boxes, e.g. the face region of the faces found by a face detector.
- struct OFIQ::LandmarkPoint

  Data structure to describe the x and y coordinate of a landmark.
- struct OFIQ::FaceLandmarks

  Data structure for storing facial landmarks.
- struct OFIQ::FaceImageQualityAssessment

  Data structure storing the results of the different measurement computations.

Namespaces

- namespace OFIQ

  Namespace for OFIQ API.

Typedefs

- using OFIQ::QualityAssessments = std::map<QualityMeasure, QualityMeasureResult>

    Data structure that stores key-value pairs, with each entry representing a quality element and its value.
- using OFIQ::Landmarks = std::vector<LandmarkPoint>

    container for a collection of landmarks, e.g. belonging to all the landmarks detected on a face image.

Enumerations

- enum class OFIQ::ReturnCode {
    OFIQ::Success = 0 , OFIQ::ImageReadingError , OFIQ::ImageWritingError , OFIQ::MissingConfigParamError
    ,
    OFIQ::UnknownConfigParamError , OFIQ::FaceDetectionError , OFIQ::FaceLandmarkExtractionError ,
    OFIQ::FaceOcclusionSegmentationError ,
    OFIQ::FaceParsingError , OFIQ::UnknownError , OFIQ::QualityAssessmentError , OFIQ::NotImplemented }

    Return codes for functions specified in this API.
- enum class OFIQ::QualityMeasure {
    OFIQ::UnifiedQualityScore = 0x41 , OFIQ::BackgroundUniformity = 0x42 , OFIQ::IlluminationUniformity =
    0x43 , OFIQ::Luminance = -0x44 ,
    OFIQ::LuminanceMean = 0x44 , OFIQ::LuminanceVariance = 0x45 , OFIQ::UnderExposurePrevention =
    0x46 , OFIQ::OverExposurePrevention = 0x47 ,
    OFIQ::DynamicRange = 0x48 , OFIQ::Sharpness = 0x49 , OFIQ::CompressionArtifacts = 0x4a ,
    OFIQ::NaturalColour = 0x4b ,
    OFIQ::SingleFacePresent = 0x4c , OFIQ::EyesOpen = 0x4d , OFIQ::MouthClosed = 0x4e , OFIQ::EyesVisible
    = 0x4f ,
    OFIQ::MouthOcclusionPrevention = 0x50 , OFIQ::FaceOcclusionPrevention = 0x51 , OFIQ::InterEyeDistance
    = 0x52 , OFIQ::HeadSize = 0x53 ,
    OFIQ::CropOfTheFaceImage = -0x54 , OFIQ::LeftwardCropOfTheFaceImage = 0x54 , OFIQ::RightwardCropOfTheFaceImage
    = 0x55 , OFIQ::DownwardCropOfTheFaceImage = 0x56 ,
    OFIQ::UpwardCropOfTheFaceImage = 0x57 , OFIQ::HeadPose = -0x58 , OFIQ::HeadPoseYaw = 0x58 ,
    OFIQ::HeadPosePitch = 0x59 ,
    OFIQ::HeadPoseRoll = 0x5a , OFIQ::ExpressionNeutrality = 0x5b , OFIQ::NoHeadCoverings = 0x5c ,
    OFIQ::NotSet = -1 }

    Enums presenting the measure labels.
- enum class OFIQ::QualityMeasureReturnCode { OFIQ::Success = 0 , OFIQ::FailureToAssess ,
    OFIQ::NotInitialized }

    Return codes for QualityMeasureResult.
- enum class OFIQ::FaceDetectorType { OFIQ::OPENCVSSD , OFIQ::NotSet }

    Enum describing the different face detector implementations.
- enum class OFIQ::LandmarkType { OFIQ::LM_98 , OFIQ::NotSet }

    Enum describing the different implementations of landmarks.

Functions

- std::ostream & OFIQ::operator<< (std::ostream &s, const ReturnCode &rc)

8.7.1 Detailed Description

PRovides several helper classes, enums and functions used in the OFIQ framework.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.8 ofiq_structs.h

[Go to the documentation of this file.](#)
```
00001
00027 #ifndef OFIQ_STRUCTS_H
00028 #define OFIQ_STRUCTS_H
00029
00030 #include <cstdint>
00031 #include <iostream>
00032 #include <map>
00033 #include <memory>
00034 #include <string>
00035 #include <utility>
00036 #include <vector>
00037
00041 namespace OFIQ
00042 {
00047     struct Image
00048     {
00049
00051         uint16_t width{ 0 };
00053         uint16_t height{ 0 };
00055         uint8_t depth{ 24 };
00060         std::shared_ptr<uint8_t> data;
00061
00065         Image() = default;
00066
00075         Image(uint16_t width, uint16_t height, uint8_t depth, std::shared_ptr<uint8_t>& data)
00076             : width{width},
00077               height{height},
00078               depth{depth},
00079               data{data}
00080         {
00081         }
00082
00084         size_t size() const { return (static_cast<size_t>(width) * height * (depth / 8)); }
00085     };
00086
00087
00092     enum class ReturnCode
00093     {
00095         Success = 0,
00097         ImageReadingError,
00099         ImageWritingError,
00101         MissingConfigParamError,
00103         UnknownConfigParamError,
00105         FaceDetectionError,
00107         FaceLandmarkExtractionError,
```

```
00109        FaceOcclusionSegmentationError,
00111        FaceParsingError,
00113        UnknownError,
00115        QualityAssessmentError,
00117        NotImplemented
00118    };
00119
00121    inline std::ostream& operator<<(std::ostream& s, const ReturnCode& rc)
00122    {
00123        switch (rc)
00124        {
00125        case ReturnCode::Success:
00126            return (s << "Success");
00127        case ReturnCode::UnknownError:
00128            return (s << "Unknown Error");
00129        case ReturnCode::QualityAssessmentError:
00130            return (s << "Failure to generate a quality score on the input image");
00131        case ReturnCode::NotImplemented:
00132            return (s << "Function is not implemented");
00133        default:
00134            return (s << "Undefined error");
00135        }
00136    }
00137
00150    struct ReturnStatus
00151    {
00153        ReturnCode code{ ReturnCode::UnknownError };
00155        std::string info;
00156
00161        ReturnStatus() = default;
00162
00172        ReturnStatus(const ReturnCode code, const std::string& info = "")
00173            : code{code},
00174              info{info}
00175        {
00176        }
00177    };
00178
00181    enum class QualityMeasure
00182    {
00184        UnifiedQualityScore = 0x41,
00186        BackgroundUniformity = 0x42,
00188        IlluminationUniformity = 0x43,
00190        Luminance = -0x44 ,
00192        LuminanceMean = 0x44,
00194        LuminanceVariance = 0x45,
00196        UnderExposurePrevention = 0x46,
00198        OverExposurePrevention = 0x47,
00200        DynamicRange = 0x48,
00202        Sharpness = 0x49,
00204        CompressionArtifacts = 0x4a ,
00206        NaturalColour = 0x4b,
00208        SingleFacePresent = 0x4c ,
00210        EyesOpen = 0x4d ,
00212        MouthClosed = 0x4e,
00214        EyesVisible = 0x4f,
00216        MouthOcclusionPrevention = 0x50,
00218        FaceOcclusionPrevention = 0x51,
00220        InterEyeDistance = 0x52,
00222        HeadSize = 0x53,
00224        CropOfTheFaceImage = -0x54,
00226        LeftwardCropOfTheFaceImage = 0x54,
00228        RightwardCropOfTheFaceImage = 0x55,
00230        DownwardCropOfTheFaceImage = 0x56,
00232        UpwardCropOfTheFaceImage = 0x57,
00234        HeadPose = -0x58,
00236        HeadPoseYaw = 0x58,
00238        HeadPosePitch = 0x59,
00240        HeadPoseRoll = 0x5a,
00242        ExpressionNeutrality = 0x5b,
00244        NoHeadCoverings = 0x5c,
00246        NotSet = -1
00247    };
00248
00253    enum class QualityMeasureReturnCode
00254    {
00256        Success = 0,
00258        FailureToAssess,
00260        NotInitialized
00261    };
00262
00267    struct QualityMeasureResult
00268    {
00270        double rawScore{ -1 };
00274        double scalar{ -1 };
00276        QualityMeasureReturnCode code{ QualityMeasureReturnCode::NotInitialized };
00277
```

```
00282          QualityMeasureResult() = default;
00283
00291          QualityMeasureResult(double rawScore, double scalar = -1, QualityMeasureReturnCode code =
      QualityMeasureReturnCode::NotInitialized)
00292               : rawScore{rawScore},
00293                 scalar{scalar},
00294                 code{code}
00295          {
00296          }
00297       };
00298
00304       using QualityAssessments = std::map<QualityMeasure, QualityMeasureResult>;
00305
00310       enum class FaceDetectorType
00311       {
00313           OPENCVSSD,
00315           NotSet
00316       };
00317
00318
00324       struct BoundingBox
00325       {
00328           int16_t xleft{ -1 };
00331           int16_t ytop{ -1 };
00333           int16_t width{ -1 };
00335           int16_t height{ -1 };
00336
00338           FaceDetectorType faceDetector = FaceDetectorType::NotSet;
00339
00344           BoundingBox() = default;
00345
00355           BoundingBox(int16_t xleft, int16_t ytop, int16_t width, int16_t height, FaceDetectorType
      i_faceDetector)
00356               : xleft{xleft},
00357                 ytop{ytop},
00358                 width{width},
00359                 height{height},
00360                 faceDetector(i_faceDetector)
00361          {
00362          }
00363       };
00364
00369       struct LandmarkPoint
00370       {
00375           int16_t x{ -1 };
00380           int16_t y{-1 };
00381
00386           LandmarkPoint() = default;
00387
00394           LandmarkPoint(int16_t i_x, int16_t i_y)
00395               : x{i_x},
00396                 y{i_y}
00397          {
00398          }
00399       };
00400
00405       using Landmarks = std::vector<LandmarkPoint>;
00406
00411       enum class LandmarkType
00412       {
00414           LM_98,
00416           NotSet
00417       };
00418
00419
00420
00425       struct FaceLandmarks
00426       {
00428           LandmarkType type{ LandmarkType::NotSet };
00430           Landmarks landmarks;
00431
00433           FaceLandmarks() = default;
00434       };
00435
00440       struct FaceImageQualityAssessment
00441       {
00442
00447           QualityAssessments qAssessments;
00448
00453           BoundingBox boundingBox;
00454
00459           FaceImageQualityAssessment() = default;
00460
00467           FaceImageQualityAssessment(
00468               const QualityAssessments& qAssessments, const BoundingBox& boundingBox)
00469               : qAssessments{qAssessments},
00470                 boundingBox{boundingBox}
```

```
00471          {
00472          }
00473     };
00474
00475 }
00476
00477 #endif /* OFIQ_STRUCTS_H */
```

8.9 AllDetectors.h File Reference

Provides the include statements to all classes derived from FaceDetectorInterface.

```
#include "opencv_ssd_face_detector.h"
```

8.9.1 Detailed Description

Provides the include statements to all classes derived from FaceDetectorInterface.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.10 AllDetectors.h

Go to the documentation of this file.
```
00001
00028 #include "opencv_ssd_face_detector.h"
```

8.11 detectors.h File Reference

Provides the interface class to the face detector implementations.

```
#include "ofiq_lib.h"
#include "Session.h"
```

Classes

- class OFIQ_LIB::FaceDetectorInterface

    Provides the interface class to the face detector implementations.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.

8.11.1 Detailed Description

Provides the interface class to the face detector implementations.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.12 detectors.h

[Go to the documentation of this file.](#)
```
00001
00028 #pragma once
00029
00030 #include "ofiq_lib.h"
00031 #include "Session.h"
00032
00036 namespace OFIQ_LIB
00037 {
00038
00043     class FaceDetectorInterface
00044     {
00045     public:
00050         virtual ~FaceDetectorInterface() = default;
00051
00058         std::vector<OFIQ::BoundingBox> detectFaces(OFIQ_LIB::Session& session);
00059
00060     protected:
00068         virtual std::vector<OFIQ::BoundingBox> UpdateFaces(OFIQ_LIB::Session& session) = 0;
00069     };
00070 }
```

8.13 opencv_ssd_face_detector.h File Reference

Implementation of a face detector using the SSD face detector CNN.

```
#include "Configuration.h"
#include "detectors.h"
#include <opencv2/dnn.hpp>
```

Classes

- class OFIQ_LIB::modules::detectors::SSDFaceDetector

    Implementation of a face detector using the SSD face detector CNN.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::detectors

    Provides face detector implementations.

8.13.1 Detailed Description

Implementation of a face detector using the SSD face detector CNN.

Author

OFIQ development team

8.14 opencv_ssd_face_detector.h

Go to the documentation of this file.
```
00001
00027 #pragma once
00028
00029 #include "Configuration.h"
00030 #include "detectors.h"
00031 #include <opencv2/dnn.hpp>
00032
00033
00037 namespace OFIQ_LIB::modules::detectors
00038 {
00039
00043     class SSDFaceDetector : public OFIQ_LIB::FaceDetectorInterface
00044     {
00045     public:
00051         explicit SSDFaceDetector(const Configuration& config);
00052
00057         ~SSDFaceDetector() override = default;
00058
00059     protected:
00067         std::vector<OFIQ::BoundingBox> UpdateFaces(OFIQ_LIB::Session& session) override;
00068
00069
00070     private:
00075         std::shared_ptr<cv::dnn::Net> dnnNet{nullptr};
00076
00081         double confidenceThreshold;
00082
00087         double padding;
00088
00093         double minimalRelativeFaceSize;
00094     };
00095 }
```

8.15 adnet_FaceMap.h File Reference

Provides definitions of landmark indices to access specific parts of ADNet landmarks.

```
#include "FaceParts.h"
#include <array>
#include <map>
#include <vector>
```

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::landmarks

    Provides implementations of a landmark extractors.
- namespace OFIQ_LIB::modules::landmarks::adnet

    Namespace for ADNet-specific landmarks.

Variables

- const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::leftEye {60,61,62,63,64,65,66,67}

    Landmark indices (ADNet) of the left eye.
- const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::rightEye {68,69,70,71,72,73,74,75}

    Landmark indices (ADNet) of the right eye.
- const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::leftEyeCorners {60,64}

    Landmark indices (ADNet) of the left eyes' corners.
- const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::rightEyeCorners {68,72}

    Landmark indices (ADNet) of the right eyes' corners.
- const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::nosetip {54}

    Landmark index (ADNet) of the nose tip.
- const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::mouthOuter {76,77,78,79,80,81,82,83,84,85,86,87}

    Landmark indices (ADNet) on the mouth's outer contour.
- const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::mouthInner {88,89,90,91,92,93,94,95}

    Landmark indices (ADNet) on the mouth's inner lip borders.
- const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::contour {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,2

    Landmark indices (ADNet) of the face contour.
- const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::forehead {}

    Landmark indices (ADNet) of the forehead (empty for ADNet).
- const LandmarkIds OFIQ_LIB::modules::landmarks::adnet::chin {16}

    Landmark index (ADNet) of the chin.
- const landmarks::FaceMap OFIQ_LIB::modules::landmarks::adnet::FaceMap

    ADNets face map definition.
- const LandmarkIdPairs OFIQ_LIB::modules::landmarks::adnet::pairsLeftEye

    Pair indices of landmarks (ADNet) for the left eye.
- const LandmarkIdPairs OFIQ_LIB::modules::landmarks::adnet::pairsRightEye

    Landmark index pairs (ADNet) of landmarks for the right eye.
- const LandmarkIdPairs OFIQ_LIB::modules::landmarks::adnet::pairsInnerLip

    Landmark index pairs (ADNet) of inner lip pairs.
- const LandmarkIdPairs OFIQ_LIB::modules::landmarks::adnet::pairsMouthCenter

    Landmark index pair (ADNet) of the inner mouth (lips) center.
- const landmarks::FacePairMap OFIQ_LIB::modules::landmarks::adnet::FacePairMap

    ADNets face pair map definition.

8.15.1 Detailed Description

Provides definitions of landmark indices to access specific parts of ADNet landmarks.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The definitions provided by this header were taken from https://arxiv.org/pdf/2109.05721.pdf Appendix A, Figure 6.

Author

OFIQ development team

8.16 adnet_FaceMap.h

Go to the documentation of this file.
```
00001
00033 #pragma once
00034
00035 #include "FaceParts.h"
00036 #include <array>
00037 #include <map>
00038 #include <vector>
00039
00043 namespace OFIQ_LIB::modules::landmarks::adnet
00044 {
00049     const LandmarkIds leftEye{60,61,62,63,64,65,66,67};
00050
00055     const LandmarkIds rightEye{68,69,70,71,72,73,74,75};
00056
00060     const LandmarkIds leftEyeCorners{60,64};
00061
00065     const LandmarkIds rightEyeCorners{68,72};
00066
00070     const LandmarkIds nosetip{54};
00071
00075     const LandmarkIds mouthOuter{76,77,78,79,80,81,82,83,84,85,86,87};
00076
00080     const LandmarkIds mouthInner{88,89,90,91,92,93,94,95};
00081
00085     const LandmarkIds
      contour{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32};
00086
00090     const LandmarkIds forehead{};
00091
00095     const LandmarkIds chin{16};
00096
00100     const landmarks::FaceMap FaceMap{
```

```
00101          {FaceParts::LEFT_EYE,          leftEye        },
00102          {FaceParts::RIGHT_EYE,         rightEye       },
00103          {FaceParts::LEFT_EYE_CORNERS,  leftEyeCorners},
00104          {FaceParts::RIGHT_EYE_CORNERS, rightEyeCorners},
00105          {FaceParts::MOUTH_OUTER,       mouthOuter     },
00106          {FaceParts::MOUTH_INNER,       mouthInner     },
00107          {FaceParts::FACE_CONTOUR,      contour        },
00108          {FaceParts::CHIN,              chin           },
00109          {FaceParts::NOSETIP,           nosetip        },
00110          {FaceParts::FOREHEAD,          forehead       }
00111      };
00112
00117      const LandmarkIdPairs pairsLeftEye{
00118          {61, 67},
00119          {62, 66},
00120          {63, 65}
00121      };
00122
00127      const LandmarkIdPairs pairsRightEye{
00128          {69, 75},
00129          {70, 74},
00130          {71, 73}
00131      };
00132
00137      const LandmarkIdPairs pairsInnerLip{
00138          {89, 95},
00139          {90, 94},
00140          {91, 93}
00141      };
00142
00147      const LandmarkIdPairs pairsMouthCenter{
00148          {90, 94}
00149      };
00150
00154      const landmarks::FacePairMap FacePairMap{
00155          {FaceParts::LEFT_EYE,      pairsLeftEye    },
00156          {FaceParts::RIGHT_EYE,     pairsRightEye   },
00157          {FaceParts::MOUTH_INNER,   pairsInnerLip   },
00158          {FaceParts::MOUTH_CENTER,  pairsMouthCenter}
00159      };
00160 }
```

8.17 adnet_landmarks.h File Reference

Provides the ADNetFaceLandmarkExtractor class.

```
#include <memory>
#include "Configuration.h"
#include "detectors.h"
#include "landmarks.h"
```

Classes

- class OFIQ_LIB::modules::landmarks::ADNetFaceLandmarkExtractor

    Class implementing the FaceLandmarkExtractorInterface interface.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::landmarks

    Provides implementations of a landmark extractors.

8.17.1 Detailed Description

Provides the ADNetFaceLandmarkExtractor class.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.18 adnet_landmarks.h

Go to the documentation of this file.
```
00001
00028 #pragma once
00029
00030 #include <memory>
00031 #include "Configuration.h"
00032 #include "detectors.h"
00033 #include "landmarks.h"
00034
00038 namespace OFIQ_LIB::modules::landmarks
00039 {
00040
00042     class ADNetFaceLandmarkExtractorImpl;
00043
00049     class ADNetFaceLandmarkExtractor : public FaceLandmarkExtractorInterface
00050     {
00051     public:
00056         explicit ADNetFaceLandmarkExtractor(const Configuration& config);
00057
00061         ~ADNetFaceLandmarkExtractor() override;
00062
00063     protected:
00071         OFIQ::FaceLandmarks updateLandmarks(OFIQ_LIB::Session& session) override;
00072
00073 #ifdef OFIQ_SINGLE_FACE_PRESENT_WITH_TMETRIC
00087         std::vector<OFIQ::FaceLandmarks> updateLandmarksAllFaces
00088         (OFIQ_LIB::Session& session, const std::vector<OFIQ::BoundingBox>& faces) override;
00089 #endif
00090
00091     private:
00092
00096         std::unique_ptr<ADNetFaceLandmarkExtractorImpl> landmarkExtractor;
00097     };
00098 }
```

8.19 AllLandmarks.h File Reference

Provides the include statements to all classes derived from FaceLandmarkExtractorInterface.

```
#include "adnet_landmarks.h"
```

8.19.1 Detailed Description

Provides the include statements to all classes derived from FaceLandmarkExtractorInterface.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.20 AllLandmarks.h

Go to the documentation of this file.
```
00001
00029 #include "adnet_landmarks.h"
```

8.21 FaceMeasures.h File Reference

Provides a class implementing two luminance measures.

```
#include "ofiq_lib.h"
#include "PartExtractor.h"
#include <opencv2/opencv.hpp>
```

Classes

- class OFIQ_LIB::modules::landmarks::FaceMeasures

    Provides static functions doing computations with landmarks.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.

- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::landmarks

    Provides implementations of a landmark extractors.

8.21.1 Detailed Description

Provides a class implementing two luminance measures.

Copyright

    Copyright (c) 2024 Federal Office for Information Security, Germany

Author

    OFIQ development team

8.22 FaceMeasures.h

Go to the documentation of this file.
```
00001
00027 #pragma once
00028
00029 #include "ofiq_lib.h"
00030 #include "PartExtractor.h"
00031 #include <opencv2/opencv.hpp>
00032
00036 namespace OFIQ_LIB::modules::landmarks
00037 {
00041     class FaceMeasures
00042     {
00043     public:
00047         FaceMeasures() = delete;
00048
00066         static double InterEyeDistance(const OFIQ::FaceLandmarks& faceLandmarks, double yaw);
00067
00078         static cv::Mat GetFaceMask
00079         (const OFIQ::FaceLandmarks& faceLandmarks, const int height, const int width,
00080          const float alpha = 0);
00081
00088         static double GetDistance(const OFIQ::LandmarkPoint& a, const OFIQ::LandmarkPoint& b);
00089
00096         static double GetDistance(const LandmarkPair& pair)
00097         {
00098             return GetDistance(pair.Lower, pair.Upper);
00099         }
00100
00106         static OFIQ::LandmarkPoint GetMiddle(const OFIQ::Landmarks& landmarks);
00107
00113         static OFIQ::LandmarkPoint GetMiddle(const LandmarkPair& pair)
00114         {
00115             return GetMiddle(OFIQ::Landmarks{pair.Lower, pair.Upper});
00116         }
00117
00125         static OFIQ::LandmarkPoint GetMiddle(const std::vector<LandmarkPair>& pairs)
00126         {
00127             std::vector<OFIQ::LandmarkPoint> points;
00128             for (auto pair : pairs)
00129             {
00130                 points.push_back(GetMiddle(pair));
00131             }
00132             return GetMiddle(points);
00133         }
00134
00147         static double GetMaxPairDistance
00148         (const OFIQ::FaceLandmarks& landmarks,
00149          landmarks::FaceParts facePart);
00150     };
00151 }
```

8.23 FaceParts.h File Reference

PRovides several helper classes, enums and functions used in the OFIQ framework.

Namespaces

- namespace OFIQ_LIB

  Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::landmarks

  Provides implementations of a landmark extractors.

Typedefs

- using OFIQ_LIB::modules::landmarks::LandmarkId = int

  Type definition of a landmark index.
- using OFIQ_LIB::modules::landmarks::LandmarkIds = std::vector<LandmarkId>

  Type definition of a list of landmark indices.
- using OFIQ_LIB::modules::landmarks::FaceMap = std::map<FaceParts, LandmarkIds>

  Type definition of a face map to access landmark indices for a queried face part.
- using OFIQ_LIB::modules::landmarks::LandmarkIdPair = std::array<LandmarkId, 2>

  Type definition for a pair of landmark index.
- using OFIQ_LIB::modules::landmarks::LandmarkIdPairs = std::vector<LandmarkIdPair>

  Type definition for a list of landmark index pairs.
- using OFIQ_LIB::modules::landmarks::FacePairMap = std::map<FaceParts, LandmarkIdPairs>

  Structure defining pairs of landmark indices.

Enumerations

- enum class OFIQ_LIB::modules::landmarks::FaceParts {
  OFIQ_LIB::modules::landmarks::LEFT_EYE , OFIQ_LIB::modules::landmarks::RIGHT_EYE , OFIQ_LIB::modules::landmarks::
  , OFIQ_LIB::modules::landmarks::RIGHT_EYE_CORNERS ,
  OFIQ_LIB::modules::landmarks::MOUTH_OUTER , OFIQ_LIB::modules::landmarks::MOUTH_INNER ,
  OFIQ_LIB::modules::landmarks::FACE_CONTOUR , OFIQ_LIB::modules::landmarks::MOUTH_CENTER
  ,
  OFIQ_LIB::modules::landmarks::CHIN , OFIQ_LIB::modules::landmarks::NOSETIP , OFIQ_LIB::modules::landmarks::FOREHE
  }

  Enumeration of facial landmark parts.

8.23.1 Detailed Description

PRovides several helper classes, enums and functions used in the OFIQ framework.

Copyright

> Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

> OFIQ development team

8.24 FaceParts.h

[Go to the documentation of this file.](#)
```
00001
00028 #pragma once
00029
00033 namespace OFIQ_LIB::modules::landmarks
00034 {
00038     enum class FaceParts
00039     {
00041         LEFT_EYE,
00043         RIGHT_EYE,
00045         LEFT_EYE_CORNERS,
00047         RIGHT_EYE_CORNERS,
00049         MOUTH_OUTER,
00051         MOUTH_INNER,
00053         FACE_CONTOUR,
00055         MOUTH_CENTER,
00057         CHIN,
00059         NOSETIP,
00061         FOREHEAD
00062     };
00063
00067     using LandmarkId = int;
00068
00072     using LandmarkIds = std::vector<LandmarkId>;
00073
00078     using FaceMap = std::map<FaceParts, LandmarkIds>;
00079
00083     using LandmarkIdPair = std::array<LandmarkId, 2>;
00084
00088     using LandmarkIdPairs = std::vector<LandmarkIdPair>;
00089
00093     using FacePairMap = std::map<FaceParts, LandmarkIdPairs>;
00094 }
```

8.25 landmarks.h File Reference

Provides the base class for the implementation of face landmark extractors.

```
#include "ofiq_lib.h"
#include "Session.h"
```

Classes

- class OFIQ_LIB::FaceLandmarkExtractorInterface

  Implements the base class for the face landmark extractors.

Namespaces

- namespace OFIQ_LIB

  Namespace for OFIQ implementations.

8.25.1 Detailed Description

Provides the base class for the implementation of face landmark extractors.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.26 landmarks.h

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

```
00001
00027 #pragma once
00028
00029 #include "ofiq_lib.h"
00030 #include "Session.h"
00031
00035 namespace OFIQ_LIB
00036 {
00041     class FaceLandmarkExtractorInterface
00042     {
00043     public:
00048         virtual ~FaceLandmarkExtractorInterface() = default;
00049
00056         OFIQ::FaceLandmarks extractLandmarks(OFIQ_LIB::Session& session);
00057 #ifdef OFIQ_SINGLE_FACE_PRESENT_WITH_TMETRIC //deprecated but required by DIS of ISO/IEC 29794-5
00058         std::vector<OFIQ::FaceLandmarks> extractLandmarksAllFaces(OFIQ_LIB::Session& session, const
    std::vector<OFIQ::BoundingBox>& faces );
00059 #endif
00060
00061     protected:
00068         virtual OFIQ::FaceLandmarks updateLandmarks(OFIQ_LIB::Session& session) = 0;
00069 #ifdef OFIQ_SINGLE_FACE_PRESENT_WITH_TMETRIC //deprecated but required by DIS of ISO/IEC 29794-5
00070         virtual std::vector<OFIQ::FaceLandmarks> updateLandmarksAllFaces(
00071             OFIQ_LIB::Session& session, const std::vector<OFIQ::BoundingBox>& faces) = 0;
00072 #endif
00073     };
00074 }
```

8.27 PartExtractor.h File Reference

Provides helper class for face landmark handling.

```
#include "ofiq_lib.h"
#include "FaceParts.h"
```

Classes

- struct OFIQ_LIB::modules::landmarks::LandmarkPair

    Data container for storing pairs of landmarks.
- class OFIQ_LIB::modules::landmarks::PartExtractor

    Class that provides helper methods for the administration of landmarks.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::landmarks

    Provides implementations of a landmark extractors.

8.27.1 Detailed Description

Provides helper class for face landmark handling.

Copyright

> Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

> OFIQ development team

8.28 PartExtractor.h

Go to the documentation of this file.
```
00001
00027 #pragma once
00028
00029 #include "ofiq_lib.h"
00030 #include "FaceParts.h"
00031
00035 namespace OFIQ_LIB::modules::landmarks
00036 {
00037
00042     enum class FaceParts;
00043
00048     struct LandmarkPair
00049     {
00054         OFIQ::LandmarkPoint Upper;
00055
00060         OFIQ::LandmarkPoint Lower;
00061
00068         LandmarkPair(OFIQ::LandmarkPoint upper, OFIQ::LandmarkPoint lower) : Upper{upper},
00069     Lower{lower}
00069         {
00070         }
00071     };
00072
00077     class PartExtractor
00078     {
00079     public:
00088         static OFIQ::Landmarks getFacePart(const OFIQ::FaceLandmarks& faceLandmarks, FaceParts part);
00089
00098         static std::vector<LandmarkPair> getPairsForPart(const OFIQ::FaceLandmarks& faceLandmarks,
00098     FaceParts part);
00099     };
00100 }
```

8.29 AllMeasures.h File Reference

Provides all classes derived from the OFIQ_LIB::modules::measures::Measure class.

```
#include "BackgroundUniformity.h"
#include "CompressionArtifacts.h"
#include "CropOfTheFaceImage.h"
#include "DynamicRange.h"
#include "ExpressionNeutrality.h"
#include "EyesOpen.h"
#include "EyesVisible.h"
#include "FaceOcclusionPrevention.h"
#include "FaceOcclusionSegmentation.h"
#include "FaceParsing.h"
#include "HeadPose.h"
#include "HeadSize.h"
#include "IlluminationUniformity.h"
#include "InterEyeDistance.h"
#include "Luminance.h"
#include "MouthClosed.h"
#include "MouthOcclusionPrevention.h"
#include "NaturalColour.h"
#include "NoHeadCoverings.h"
#include "OverExposurePrevention.h"
#include "Sharpness.h"
#include "SingleFacePresent.h"
#include "UnderExposurePrevention.h"
#include "UnifiedQualityScore.h"
```

8.29.1 Detailed Description

Provides all classes derived from the OFIQ_LIB::modules::measures::Measure class.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.30 AllMeasures.h

Go to the documentation of this file.
```
00001
00029 #include "BackgroundUniformity.h"
00030 #include "CompressionArtifacts.h"
00031 #include "CropOfTheFaceImage.h"
00032 #include "DynamicRange.h"
00033 #include "ExpressionNeutrality.h"
00034 #include "EyesOpen.h"
00035 #include "EyesVisible.h"
00036 #include "FaceOcclusionPrevention.h"
00037 #include "FaceOcclusionSegmentation.h"
00038 #include "FaceParsing.h"
00039 #include "HeadPose.h"
00040 #include "HeadSize.h"
00041 #include "IlluminationUniformity.h"
00042 #include "InterEyeDistance.h"
00043 #include "Luminance.h"
00044 #include "MouthClosed.h"
00045 #include "MouthOcclusionPrevention.h"
00046 #include "NaturalColour.h"
00047 #include "NoHeadCoverings.h"
00048 #include "OverExposurePrevention.h"
00049 #include "Sharpness.h"
00050 #include "SingleFacePresent.h"
00051 #include "UnderExposurePrevention.h"
00052 #include "UnifiedQualityScore.h"
```

8.31 BackgroundUniformity.h File Reference

Provides a class implementing the background uniformity measure.

```
#include "Measure.h"
```

Classes

- class OFIQ_LIB::modules::measures::BackgroundUniformity

    Implementation of the background uniformity measure.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

    Provides measures implemented in OFIQ.

8.31.1 Detailed Description

Provides a class implementing the background uniformity measure.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Author

OFIQ development team

8.32 BackgroundUniformity.h

[Go to the documentation of this file.](#)
```
00001
00028 #pragma once
00029
00030 #include "Measure.h"
00031
00035 namespace OFIQ_LIB::modules::measures
00036 {
00043     class BackgroundUniformity : public Measure
00044     {
00045     public:
00053         BackgroundUniformity(
00054             const Configuration& configuration,
00055             Session& session);
00056
00064         void Execute(OFIQ_LIB::Session & session) override;
00065
00066     private:
00071         uint16_t m_target_height = 292;
00072
00077         uint16_t m_target_width = 354;
00078
00084         uint16_t m_crop_left = 62;
00085
00091         uint16_t m_crop_right = 62;
00092
00098         uint16_t m_crop_top = 0;
00099
00105         uint16_t m_crop_bottom = 210;
00106
00113         uint16_t m_erosion_kernel_size = 4;
00114     };
00115 }
```

8.33 CompressionArtifacts.h File Reference

Provides a class implemtenting the no compression artifact measure.

```
#include "landmarks.h"
#include "Measure.h"
#include <ONNXRTSegmentation.h>
```

Classes

- class OFIQ_LIB::modules::measures::CompressionArtifacts

  Implementation of the no compression artifacts measure.

Namespaces

- namespace OFIQ_LIB

  Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

  Provides measures implemented in OFIQ.

8.33.1 Detailed Description

Provides a class implemtenting the no compression artifact measure.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.34 CompressionArtifacts.h

Go to the documentation of this file.
```
00001
00028 #pragma once
00029
00030 #include "landmarks.h"
00031 #include "Measure.h"
00032 #include <ONNXRTSegmentation.h>
00033
00037 namespace OFIQ_LIB::modules::measures
00038 {
00045     class CompressionArtifacts : public Measure
00046     {
00047     public:
00071         CompressionArtifacts(const Configuration& configuration, Session& session);
00072
00081         void Execute(OFIQ_LIB::Session& session) override;
00082
00083     private:
00090         uint16_t m_crop;
00091
00099         uint16_t m_dim;
00100
00104         ONNXRuntimeSegmentation m_onnxRuntimeEnv;
00105     };
00106 }
```

8.35 CropOfTheFaceImage.h File Reference

Provides a class implementing the crop of the face image measure.

```
#include "landmarks.h"
#include "Measure.h"
```

Classes

- class OFIQ_LIB::modules::measures::CropOfTheFaceImage

  Implementation of the crop of the face image measure.

Namespaces

- namespace OFIQ_LIB

  Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

  Provides measures implemented in OFIQ.

8.35.1 Detailed Description

Provides a class implementing the crop of the face image measure.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ devlopment team

8.36 CropOfTheFaceImage.h

Go to the documentation of this file.
```
00001
00027 #pragma once
00028
00029 #include "landmarks.h"
00030 #include "Measure.h"
00031
00035 namespace OFIQ_LIB::modules::measures
00036 {
00042     class CropOfTheFaceImage : public Measure
00043     {
00044     public:
00052         CropOfTheFaceImage(const Configuration& configuration, Session& session);
00053
00060         void Execute(OFIQ_LIB::Session & session) override;
00061     };
00062 }
```

8.37 DynamicRange.h File Reference

Provides a class implemtenting the dynamic range measure.

```
#include "landmarks.h"
#include "Measure.h"
```

Classes

- class OFIQ_LIB::modules::measures::DynamicRange

  Implementation of the dynamic range measure.

Namespaces

- namespace OFIQ_LIB

  Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

  Provides measures implemented in OFIQ.

8.37.1 Detailed Description

Provides a class implemtenting the dynamic range measure.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.38 DynamicRange.h

[Go to the documentation of this file.](#)
```
00001
00028 #pragma once
00029
00030 #include "landmarks.h"
00031 #include "Measure.h"
00032
00036 namespace OFIQ_LIB::modules::measures
00037 {
00043     class DynamicRange : public Measure
00044     {
00045     public:
00053         DynamicRange(
00054             const Configuration& configuration,
00055             Session& session);
00056
00062         void Execute(OFIQ_LIB::Session & session) override;
00063     };
00064 }
```

8.39 Executor.h File Reference

This class takes care of the computation of the measures activated.

```
#include "Measure.h"
```

Classes

- class OFIQ_LIB::modules::measures::Executor

    This class takes care of the computation of the measures activated.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

    Provides measures implemented in OFIQ.

Functions

- void OFIQ_LIB::modules::measures::log (const std::string_view &msg)

    Logging function for writing debug messages to std::cout.

Variables

- static const bool OFIQ_LIB::modules::measures::execLogActive = false

    This variable enables logging to std::cout for debug purposes. By default the logging is switched off.

8.39.1 Detailed Description

This class takes care of the computation of the measures activated.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Author

OFIQ development team

8.40 Executor.h

Go to the documentation of this file.

```
00001
00027 #pragma once
00028
00029 #include "Measure.h"
00030
00034 namespace OFIQ_LIB::modules::measures
00035 {
00039     const static bool execLogActive = false;
00040
00046     void log(const std::string_view& msg);
00047
00051     class Executor
00052     {
00053     public:
00059         explicit Executor(std::vector<std::unique_ptr<Measure» measures)
00060             : measures{std::move(measures)}
00061         {
00062         }
00063
00069         void ExecuteAll(Session & i_currentSession) const;
00070
00075         const std::vector<std::unique_ptr<Measure»& GetMeasures() const { return measures; }
00076
00077     private:
00082         std::vector<std::unique_ptr<Measure» measures;
00083     };
00084 }
```

8.41 ExpressionNeutrality.h File Reference

Provides a class implementing the expression neutrality measure.

```
#include "landmarks.h"
#include "Measure.h"
#include <ONNXRTSegmentation.h>
```

Classes

- class OFIQ_LIB::modules::measures::ExpressionNeutrality

    Provides a class implementing the expression neutrality measure.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.

- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

    Provides measures implemented in OFIQ.

8.41.1 Detailed Description

Provides a class implementing the expression neutrality measure.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.42 ExpressionNeutrality.h

Go to the documentation of this file.
```
00001
00028 #pragma once
00029
00030 #include "landmarks.h"
00031 #include "Measure.h"
00032 #include <ONNXRTSegmentation.h>
00033
00037 namespace OFIQ_LIB::modules::measures
00038 {
00047     class ExpressionNeutrality : public Measure
00048     {
00049     public:
00057         ExpressionNeutrality(
00058             const Configuration& configuration, Session& session);
00059
00065         void Execute(OFIQ_LIB::Session& session) override;
00066
00067     private:
00072         ONNXRuntimeSegmentation m_onnxRuntimeEnvCNN1;
00073
00078         ONNXRuntimeSegmentation m_onnxRuntimeEnvCNN2;
00079
00084         std::shared_ptr<cv::ml::Boost> classifier;
00085     };
00086 }
```

8.43   EyesOpen.h File Reference

Provides a class implementing the eyes open measure.

```
#include "landmarks.h"
#include "Measure.h"
```

Classes

- class OFIQ_LIB::modules::measures::EyesOpen

    Implementation of the eyes open measure.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

    Provides measures implemented in OFIQ.

8.43.1   Detailed Description

Provides a class implementing the eyes open measure.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Author

OFIQ development team

8.44 EyesOpen.h

```
00001
00028 #pragma once
00029
00030 #include "landmarks.h"
00031 #include "Measure.h"
00032
00036 namespace OFIQ_LIB::modules::measures
00037 {
00043     class EyesOpen : public Measure
00044     {
00045     public:
00053         EyesOpen(const Configuration& configuration, Session& session);
00054
00063         void Execute(OFIQ_LIB::Session & session) override;
00064     };
00065 }
```

8.45 EyesVisible.h File Reference

Provides a class implementing the eyes visible measure.

```
#include "landmarks.h"
#include "Measure.h"
```

Classes

- class OFIQ_LIB::modules::measures::EyesVisible

    Implementation of the eyes visible measure.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

    Provides measures implemented in OFIQ.

8.45.1 Detailed Description

Provides a class implementing the eyes visible measure.

Author

OFIQ development team

8.46 EyesVisible.h

Go to the documentation of this file.
```
00001
00027 #pragma once
00028
00029 #include "landmarks.h"
00030 #include "Measure.h"
00031
00035 namespace OFIQ_LIB::modules::measures
00036 {
00043     class EyesVisible : public Measure
00044     {
00045     public:
00053         EyesVisible(const Configuration& configuration, Session& session);
00054
00064         void Execute(OFIQ_LIB::Session & session) override;
00065     };
00066 }
```

8.47 FaceOcclusionPrevention.h File Reference

Provides a class implementing the face occlusion prevention measure.

```
#include "landmarks.h"
#include "Measure.h"
#include <ONNXRTSegmentation.h>
```

Classes

- class OFIQ_LIB::modules::measures::FaceOcclusionPrevention

    Implementation of the face occlusion prevention measure.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

    Provides measures implemented in OFIQ.

8.47.1 Detailed Description

Provides a class implementing the face occlusion prevention measure.

Copyright

   Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

   OFIQ development team

8.48 FaceOcclusionPrevention.h

Go to the documentation of this file.
```
00001
00027 #pragma once
00028
00029 #include "landmarks.h"
00030 #include "Measure.h"
00031 #include <ONNXRTSegmentation.h>
00032
00036 namespace OFIQ_LIB::modules::measures
00037 {
00044     class FaceOcclusionPrevention : public Measure
00045     {
00046     public:
00054         FaceOcclusionPrevention(
00055             const Configuration& configuration,
00056             Session& session);
00057
00068         void Execute(OFIQ_LIB::Session & session) override;
00069     };
00070 }
```

8.49 HeadPose.h File Reference

Provides a class implementing head pose measures.

```
#include "Measure.h"
```

Classes

- class OFIQ_LIB::modules::measures::HeadPose

    Implementation of head pose measures.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

    Provides measures implemented in OFIQ.

8.49.1 Detailed Description

Provides a class implementing head pose measures.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.50 HeadPose.h

[Go to the documentation of this file.](#)
```
00001
00027 #pragma once
00028
00029 #include "Measure.h"
00030
00034 namespace OFIQ_LIB::modules::measures
00035 {
00040     class HeadPose : public Measure
00041     {
00042     public:
00050         HeadPose(
00051             const Configuration& configuration,
00052             Session& session);
00053
00061         void Execute(OFIQ_LIB::Session & session) override;
00062     };
00063 }
```

8.51 HeadSize.h File Reference

Provides a class implementing the head size measure.

```
#include "landmarks.h"
#include "Measure.h"
```

Classes

- class [OFIQ_LIB::modules::measures::HeadSize](#)

    Implementation of the head size measure.

Namespaces

- namespace [OFIQ_LIB](#)

    Namespace for OFIQ implementations.
- namespace [OFIQ_LIB::modules](#)
- namespace [OFIQ_LIB::modules::measures](#)

    Provides measures implemented in OFIQ.

8.51.1 Detailed Description

Provides a class implementing the head size measure.

Author

OFIQ development team

8.52 HeadSize.h

Go to the documentation of this file.
```
00001
00027 #pragma once
00028
00029 #include "landmarks.h"
00030 #include "Measure.h"
00031
00035 namespace OFIQ_LIB::modules::measures
00036 {
00042     class HeadSize : public Measure
00043     {
00044     public:
00052         HeadSize(
00053             const Configuration& configuration,
00054             Session& session);
00055
00061         void Execute(OFIQ_LIB::Session & session) override;
00062     };
00063 }
```

8.53 IlluminationUniformity.h File Reference

Provides a class implementing the illumination uniformity measure.

```
#include "landmarks.h"
#include "Measure.h"
```

Classes

- class OFIQ_LIB::modules::measures::IlluminationUniformity

    Implementation of the illumination uniformity measure.

Namespaces

- namespace OFIQ_LIB

  Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

  Provides measures implemented in OFIQ.

8.53.1 Detailed Description

Provides a class implementing the illumination uniformity measure.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN- CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.54 IlluminationUniformity.h

Go to the documentation of this file.
```cpp
00001
00028 #pragma once
00029
00030 #include "landmarks.h"
00031 #include "Measure.h"
00032
00036 namespace OFIQ_LIB::modules::measures
00037 {
00044     class IlluminationUniformity : public Measure
00045     {
00046     public:
00054         IlluminationUniformity(const Configuration& configuration, Session& session);
00055
00064         void Execute(OFIQ_LIB::Session & session) override;
00065     };
00066 }
```

8.55 InterEyeDistance.h File Reference

Provides a class implementing the inter-eye distance measure.

```
#include "landmarks.h"
#include "Measure.h"
```

Classes

- class OFIQ_LIB::modules::measures::InterEyeDistance

    Implementation of the inter-eye distance measure.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

    Provides measures implemented in OFIQ.

8.55.1 Detailed Description

Provides a class implementing the inter-eye distance measure.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.56 InterEyeDistance.h

[Go to the documentation of this file.](#)
```
00001
00027 #pragma once
00028
00029 #include "landmarks.h"
00030 #include "Measure.h"
00031
00035 namespace OFIQ_LIB::modules::measures
00036 {
00043     class InterEyeDistance : public Measure
00044     {
00045     public:
00053         InterEyeDistance(const Configuration& configuration, Session& session);
00054
00063         void Execute(OFIQ_LIB::Session & session) override;
00064     };
00065 }
```

8.57 Luminance.h File Reference

Provides a class implementing two luminance measures.

```
#include "landmarks.h"
#include "Measure.h"
```

Classes

- class OFIQ_LIB::modules::measures::Luminance

  Implementation of two luminance measures.

Namespaces

- namespace OFIQ_LIB

  Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

  Provides measures implemented in OFIQ.

8.57.1 Detailed Description

Provides a class implementing two luminance measures.

Author

OFIQ development team

8.58 Luminance.h

Go to the documentation of this file.
```
00001
00027 #pragma once
00028
00029 #include "landmarks.h"
00030 #include "Measure.h"
00031
00035 namespace OFIQ_LIB::modules::measures
00036 {
00042     class Luminance : public Measure
00043     {
00044     public:
00052         Luminance(const Configuration& configuration, Session& session);
00053
00061         void Execute(OFIQ_LIB::Session & session) override;
00062     };
00063 }
```

8.59 Measure.h File Reference

Provides the base class for all measures implemented in OFIQ.

```
#include "Configuration.h"
#include "ofiq_lib.h"
#include "Session.h"
#include <math.h>
```

Classes

- struct OFIQ_LIB::modules::measures::SigmoidParameters

    Parameters of the sigmoid function based quality mapping.

- class OFIQ_LIB::modules::measures::Measure

    Base class for measures implemented in OFIQ.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.

- namespace OFIQ_LIB::modules

- namespace OFIQ_LIB::modules::measures

    Provides measures implemented in OFIQ.

8.59.1 Detailed Description

Provides the base class for all measures implemented in OFIQ.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.60 Measure.h

Go to the documentation of this file.
```
00001
00027 #pragma once
00028
00029 #include "Configuration.h"
00030 #include "ofiq_lib.h"
00031 #include "Session.h"
00032 #ifndef _WIN32
00033 #    include <math.h>
00034 #endif
00035
00039 namespace OFIQ_LIB::modules::measures
00040 {
00056     struct SigmoidParameters
00057     {
00063         SigmoidParameters() { Reset(); }
00064
00069         double h;
00070
00075         double a;
00076
00081         double s;
00082
```

```
00088          double x0;
00089
00095          double w;
00096
00102          bool round;
00103
00112          void setInverse()
00113          {
00114              a = 1;
00115              s = -1;
00116          }
00117
00121          void Reset()
00122          {
00123              h = 100;
00124              a = 0;
00125              s = 1;
00126              x0 = 4;
00127              w = 0.7;
00128              round = true;
00129          }
00130      };
00131
00135      class Measure
00136      {
00137      public:
00146          Measure(const Configuration& configuration,
00147              const OFIQ_LIB::Session&,
00148              OFIQ::QualityMeasure measure)
00149              : configuration{configuration}, m_measure(measure)
00150          {
00151          }
00152
00162          virtual void Execute(OFIQ_LIB::Session& session) = 0;
00163
00167          virtual ~Measure() = default;
00168
00176          virtual std::string GetName() const;
00177
00182          virtual OFIQ::QualityMeasure GetQualityMeasure() const;
00183
00197          void SetQualityMeasure(OFIQ_LIB::Session& session, OFIQ::QualityMeasure measure, double
      rawValue, OFIQ::QualityMeasureReturnCode code);
00198
00199      protected:
00207          static double Sigmoid(double x, double x0, double w)
00208          {
00209              return 1.0 / (1 + exp((x0 - x) / w));
00210          }
00211
00224          void AddSigmoid(OFIQ::QualityMeasure measure, const SigmoidParameters& defaultValues);
00225
00238          void AddSigmoid(const std::string& key, SigmoidParameters defaultValues);
00239
00247          double ExecuteScalarConversion(OFIQ::QualityMeasure measure, double rawValue);
00248
00256          double ExecuteScalarConversion(const std::string& key, double rawValue);
00257
00262          const Configuration& configuration;
00263
00264      private:
00275          static double ScalarConversion(double rawValue, const SigmoidParameters& par)
00276          {
00277              double scalarScore = par.h * (par.a + par.s * Sigmoid(rawValue, par.x0, par.w));
00278              if (par.round)
00279                  scalarScore = round(scalarScore);
00280              if (scalarScore < 0.0)
00281              {
00282                  scalarScore = 0.0;
00283              }
00284              else if (scalarScore > 100.0)
00285              {
00286                  scalarScore = 100.0;
00287              }
00288              return scalarScore;
00289          }
00290
00295          std::map<std::string, SigmoidParameters, std::less<» sigmoidMap;
00296
00302          static std::string GetMeasureName(OFIQ::QualityMeasure measure);
00303
00311          static std::string ExpandKey(std::string_view rawKey);
00312
00318          OFIQ::QualityMeasure m_measure = OFIQ::QualityMeasure::NotSet;
00319      };
00320 }
```

8.61 MeasureFactory.h File Reference

Provides a class for requesting creation of measure implementations.

```
#include "Configuration.h"
#include "ofiq_lib.h"
#include "Measure.h"
#include "Session.h"
```

Classes

- class OFIQ_LIB::modules::measures::MeasureFactory

  Measure factor class.

Namespaces

- namespace OFIQ_LIB

  Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

  Provides measures implemented in OFIQ.

8.61.1 Detailed Description

Provides a class for requesting creation of measure implementations.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.62 MeasureFactory.h

[Go to the documentation of this file.](#)
```
00001
00028 #pragma once
00029
00030 #include "Configuration.h"
00031 #include "ofiq_lib.h"
00032 #include "Measure.h"
00033 #include "Session.h"
00034
00038 namespace OFIQ_LIB::modules::measures
00039 {
00043     class MeasureFactory
00044     {
00045      public:
00046         // Avoids instantiation from this class
00047         MeasureFactory() = delete;
00048
00063         static std::unique_ptr<Measure> CreateMeasure(
00064             const OFIQ::QualityMeasure measure,
00065             const Configuration& configuration,
00066             OFIQ_LIB::Session& session);
00067     };
00068 }
```

8.63 MouthClosed.h File Reference

Provides a class implementing the mouth closed measure.

```
#include "landmarks.h"
#include "Measure.h"
```

Classes

- class OFIQ_LIB::modules::measures::MouthClosed

    Implementation of the mouth closed measure.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.

- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

    Provides measures implemented in OFIQ.

8.63.1 Detailed Description

Provides a class implementing the mouth closed measure.

Copyright

> Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

> OFIQ development team

8.64 MouthClosed.h

[Go to the documentation of this file.](#)
```
00001
00027 #pragma once
00028
00029 #include "landmarks.h"
00030 #include "Measure.h"
00031
00035 namespace OFIQ_LIB::modules::measures
00036 {
00042     class MouthClosed : public Measure
00043     {
00044     public:
00052         MouthClosed(
00053             const Configuration& configuration, Session& session);
00054
00063         void Execute(OFIQ_LIB::Session& session) override;
00064     };
00065 }
```

8.65 MouthOcclusionPrevention.h File Reference

Provides a class implementing the mouth occlusion prevention measure.

```
#include "Measure.h"
```

Classes

- class [OFIQ_LIB::modules::measures::MouthOcclusionPrevention](#)

  Implementation of the mouth occlusion prevention measure.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.

- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

    Provides measures implemented in OFIQ.

8.65.1 Detailed Description

Provides a class implementing the mouth occlusion prevention measure.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Author

OFIQ development team

8.66 MouthOcclusionPrevention.h

Go to the documentation of this file.
```
00001
00027 #pragma once
00028
00029 #include "Measure.h"
00030
00034 namespace OFIQ_LIB::modules::measures
00035 {
00042     class MouthOcclusionPrevention : public Measure
00043     {
00044     public:
00052         MouthOcclusionPrevention(const Configuration& configuration, Session& session);
00053
00065         void Execute(OFIQ_LIB::Session & session) override;
00066     };
00067 }
```

8.67 NaturalColour.h File Reference

Provides a class implementing the natural colour measure.

```
#include "landmarks.h"
#include "Measure.h"
```

Classes

- class OFIQ_LIB::modules::measures::NaturalColour

  Implementation of the natural colour measure.

Namespaces

- namespace OFIQ_LIB

  Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

  Provides measures implemented in OFIQ.

8.67.1 Detailed Description

Provides a class implementing the natural colour measure.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.68 NaturalColour.h

```
00001
00027 #pragma once
00028
00029 #include "landmarks.h"
00030 #include "Measure.h"
00031
00035 namespace OFIQ_LIB::modules::measures
00036 {
00043     class NaturalColour : public Measure
00044     {
00045     public:
00053         NaturalColour(
00054             const Configuration& configuration,
00055             Session& session);
00056
00065         void Execute(OFIQ_LIB::Session & session) override;
00066
00067     private:
00074         cv::Mat CreateMaskedImage(const OFIQ::FaceLandmarks& landmarks, const cv::Mat& cvImage);
00075
00086         cv::Mat ReduceImageToRegionOfInterest
00087         (const cv::Mat maskedImage,
00088          const cv::Rect& leftRegionOfInterest,
00089          const cv::Rect& rightRegionOfInterest);
00090
00104         double CalculateScore(double meanChannelA, double meanChannelB);
00105     };
00106 }
```

8.69 NoHeadCoverings.h File Reference

Provides a class implementing the no head covering measure.

```
#include "Measure.h"
#include "segmentations.h"
```

Classes

- class OFIQ_LIB::modules::measures::NoHeadCoverings

    Implementation of the no head covering measure.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

    Provides measures implemented in OFIQ.

8.69.1 Detailed Description

Provides a class implementing the no head covering measure.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.70 NoHeadCoverings.h

[Go to the documentation of this file.](#)
```
00001
00027 #pragma once
00028
00029 #include "Measure.h"
00030 #include "segmentations.h"
00031
00035 namespace OFIQ_LIB::modules::measures
00036 {
00051     class NoHeadCoverings : public Measure
00052     {
00053     public:
00066         NoHeadCoverings(
00067             const Configuration& configuration,
00068             Session& session);
00069
00086         void Execute(OFIQ_LIB::Session & session) override;
00087
00088     private:
00096         double threshold;
00097     };
00098 }
```

8.71 OverExposurePrevention.h File Reference

Provides a class implementing the background uniformity measure.

```
#include "Measure.h"
```

Classes

- class OFIQ_LIB::modules::measures::OverExposurePrevention

    Implementation of the over-exposure prevention measure.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.

- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

    Provides measures implemented in OFIQ.

8.71.1 Detailed Description

Provides a class implementing the background uniformity measure.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Author

OFIQ development team

8.72 OverExposurePrevention.h

Go to the documentation of this file.
```
00001
00027 #pragma once
00028
00029 #include "Measure.h"
00030
00031
00035 namespace OFIQ_LIB::modules::measures
00036 {
00042     class OverExposurePrevention : public Measure
00043     {
00044     public:
00052         OverExposurePrevention(
00053             const Configuration& configuration,
00054             Session& session);
00055
00061         void Execute(OFIQ_LIB::Session & session) override;
00062     };
00063 }
```

8.73 Sharpness.h File Reference

Provides a class implementing the sharpness measure.

```
#include "Measure.h"
```

Classes

- class OFIQ_LIB::modules::measures::Sharpness

  Implemantation of the sharpness measure.

Namespaces

- namespace OFIQ_LIB

  Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

  Provides measures implemented in OFIQ.

8.73.1 Detailed Description

Provides a class implementing the sharpness measure.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.74   Sharpness.h

Go to the documentation of this file.
```
00001
00027 #pragma once
00028
00029 #include "Measure.h"
00030
00034 namespace OFIQ_LIB::modules::measures
00035 {
00043     class Sharpness : public Measure
00044     {
00045     public:
00053         Sharpness(
00054             const Configuration& configuration,
00055             Session& session);
00056
00062         void Execute(OFIQ_LIB::Session & session) override;
00063
00064     private:
00065
00069         std::string modelFile;
00070
00075         std::shared_ptr<cv::ml::RTrees> rtree;
00076
00082         bool useAligned;
00083
00089         double faceRegionAlpha;
00090
00095         int numTrees;
00096
00106         void GetCroppedImages(Session& session, cv::Mat& faceCrop, cv::Mat& maskCrop, bool useAligned,
    float faceRegionAlpha);
00107
00116         cv::Mat GetClassifierFocusFeatures(cv::Mat& image, cv::Mat& mask, bool applyBlur);
00117     };
00118 }
```

8.75   SingleFacePresent.h File Reference

Provides a class implementing the single face present measure.

```
#include "detectors.h"
#include "Measure.h"
```

Classes

- class OFIQ_LIB::modules::measures::SingleFacePresent

  Implementation of the single face present measure.

Namespaces

- namespace OFIQ_LIB

  Namespace for OFIQ implementations.

- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

  Provides measures implemented in OFIQ.

8.75.1 Detailed Description

Provides a class implementing the single face present measure.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.76 SingleFacePresent.h

Go to the documentation of this file.
```
00001
00027 #pragma once
00028
00029 #include "detectors.h"
00030 #include "Measure.h"
00031
00035 namespace OFIQ_LIB::modules::measures
00036 {
00041     class SingleFacePresent : public Measure
00042     {
00043     public:
00051         SingleFacePresent(
00052             const Configuration& configuration, Session& session);
00053
00060         void Execute(OFIQ_LIB::Session & session) override;
00061     };
00062 }
```

8.77 UnderExposurePrevention.h File Reference

Provides a class implemtenting the under-exposure prevention measure.

```
#include "Measure.h"
```

Classes

- class OFIQ_LIB::modules::measures::UnderExposurePrevention

    Implementation of the under-exposure prevention measure.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

    Provides measures implemented in OFIQ.

8.77.1 Detailed Description

Provides a class implemtenting the under-exposure prevention measure.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Author

OFIQ development team

8.78 UnderExposurePrevention.h

Go to the documentation of this file.
```
00001
00027 #pragma once
00028
00029 #include "Measure.h"
00030
00034 namespace OFIQ_LIB::modules::measures
00035 {
00041     class UnderExposurePrevention : public Measure
00042     {
00043     public:
00051         UnderExposurePrevention(
00052             const Configuration& configuration,
00053             Session& session);
00054
00060         void Execute(OFIQ_LIB::Session & session) override;
00061     };
00062 }
```

8.79 UnifiedQualityScore.h File Reference

Provides a class implemtenting the unified quality measure.

```
#include "landmarks.h"
#include "Measure.h"
#include <opencv2/dnn.hpp>
#include <ONNXRTSegmentation.h>
```

Classes

- class OFIQ_LIB::modules::measures::UnifiedQualityScore

    Implementation of the unified quality measure.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::measures

    Provides measures implemented in OFIQ.

8.79.1 Detailed Description

Provides a class implemtenting the unified quality measure.

Copyright

    Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

    OFIQ development team

8.80 UnifiedQualityScore.h

[Go to the documentation of this file.](#)
```
00001
00027 #pragma once
00028
00029 #include "landmarks.h"
00030 #include "Measure.h"
00031 #include <opencv2/dnn.hpp>
00032 #include <ONNXRTSegmentation.h>
00033
00037 namespace OFIQ_LIB::modules::measures
00038 {
00046     class UnifiedQualityScore : public Measure
00047     {
00048     public:
00056         UnifiedQualityScore(
00057             const Configuration& configuration,
00058             Session& session);
00059
00069         void Execute(OFIQ_LIB::Session & session) override;
00070
00071     private:
00076         ONNXRuntimeSegmentation m_onnxRuntimeEnv;
00077     };
00078 }
```

8.81 AllPoseEstimators.h File Reference

```
#include "HeadPose3DDFAV2.h"
```

8.81.1 Detailed Description

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

@briefPProvides the include statements to all classes derived from PoseEstimatorInterface.

Author

OFIQ development team

8.82 AllPoseEstimators.h

Go to the documentation of this file.
```
00001
00029 #include "HeadPose3DDFAV2.h"
```

8.83 HeadPose3DDFAV2.h File Reference

Provides a class implementing a head pose estimator based on https://github.com/cleardusk/3←
DDFA_V2.

```
#include "Configuration.h"
#include "poseEstimators.h"
#include <onnxruntime_cxx_api.h>
#include <opencv2/core/mat.hpp>
```

Classes

- class OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2

  Implementation of a head pose estimator.

Namespaces

- namespace OFIQ_LIB

  Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::poseEstimators

  Provides implementation of a head pose estimator.

8.83.1 Detailed Description

Provides a class implementing a head pose estimator based on https://github.com/cleardusk/3←
DDFA_V2.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.84 HeadPose3DDFAV2.h

[Go to the documentation of this file.](#)
```
00001
00027 #pragma once
00028
00029 #include "Configuration.h"
00030 #include "poseEstimators.h"
00031 #include <onnxruntime_cxx_api.h>
00032 #include <opencv2/core/mat.hpp>
00033
00038 namespace OFIQ_LIB::modules::poseEstimators
00039 {
00044     class HeadPose3DDFAV2 : public PoseEstimatorInterface
00045     {
00046     public:
00053         explicit HeadPose3DDFAV2(const Configuration& config);
00054
00058         ~HeadPose3DDFAV2() override = default;
00059
00060     protected:
00068         void updatePose(OFIQ_LIB::Session& session, EulerAngle& pose) override;
00069
00070     private:
00074         static const std::string paramPoseEstimatorModel;
00075
00079         Ort::Env m_ortenv;
00080
00084         std::unique_ptr<Ort::Session> m_ort_session;
00085
00089         int64_t m_expected_image_width = 0;
00090
00094         int64_t m_expected_image_height = 0;
00095
00099         int64_t m_expected_image_number_of_channels = 0;
00100
00104         int64_t m_number_of_input_elements = 0;
00105
00109         std::array<int64_t, 4> inputShape;
00110
00118         cv::Mat CropImage(const cv::Mat& image, const OFIQ::BoundingBox& biggestFace);
00119     };
00120 }
```

8.85 poseEstimators.h File Reference

Base class for the different implementation of pose estimation algorithms.

```
#include "ofiq_lib.h"
#include "Session.h"
#include <array>
```

Classes

- class OFIQ_LIB::PoseEstimatorInterface

  Implementation of the base class for integrating pose estimation algorithms capable of estimating three head orientation angles (yaw, pitch and roll) from a face image.

Namespaces

- namespace OFIQ_LIB

  Namespace for OFIQ implementations.

8.85.1 Detailed Description

Base class for the different implementation of pose estimation algorithms.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.86 poseEstimators.h

Go to the documentation of this file.
```
00001
00027 #pragma once
00028
00029 #include "ofiq_lib.h"
00030 #include "Session.h"
00031 #include <array>
00032
00036 namespace OFIQ_LIB
00037 {
00038
00043     class PoseEstimatorInterface
00044     {
00045     public:
00046         using EulerAngle = std::array<double, 3>;
00047
00052         virtual ~PoseEstimatorInterface() = default;
00053
00061         EulerAngle& estimatePose(OFIQ_LIB::Session& session);
00062
00063     protected:
00070         virtual void updatePose(OFIQ_LIB::Session& session, EulerAngle& pose) = 0;
00071
00072     private:
00077         std::string lastSessionId;
00078
00083         EulerAngle pose;
00084     };
00085 }
```

8.87 FaceOcclusionSegmentation.h File Reference

Provides a class for segmenting the facial part not occluded by any non-facial parts from an image.

```
#include "Configuration.h"
#include "segmentations.h"
#include <ONNXRTSegmentation.h>
```

Classes

- class OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation

  Class managing the separation of facial parts not occluded by non-facial parts from other parts.

Namespaces

- namespace cv

  OpenCV's namespace.
- namespace OFIQ_LIB

  Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::segmentations

  Provides segmentation-related implementations.

8.87.1 Detailed Description

Provides a class for segmenting the facial part not occluded by any non-facial parts from an image.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.88 FaceOcclusionSegmentation.h

Go to the documentation of this file.
```
00001
00028 #pragma once
00029
00030 #include "Configuration.h"
00031 #include "segmentations.h"
00032 #include <ONNXRTSegmentation.h>
00033
00037 namespace cv
00038 {
00042     class Mat;
00043 }
00044
00048 namespace OFIQ_LIB::modules::segmentations
00049 {
00056     class FaceOcclusionSegmentation : public SegmentationExtractorInterface
00057     {
00058     public:
00065         explicit FaceOcclusionSegmentation(const Configuration& config);
00066
00070         ~FaceOcclusionSegmentation() override = default;
00071
00072
00073     protected:
00093         OFIQ::Image UpdateMask(
00094             OFIQ_LIB::Session& session, modules::segmentations::SegmentClassLabels faceSegment)
       override;
00095
00096     private:
00097
00105         cv::Mat GetFaceOcclusionSegmentation(const cv::Mat& alignedImage);
00106
00110         ONNXRuntimeSegmentation m_onnxRuntimeEnv;
00111
00117         std::shared_ptr<cv::Mat> segmentationImage;
00118
00123         const std::string modelConfigItem = "params.measures.FaceOcclusionSegmentation.model_path";
00124
00128         const int cropLeft = 96;
00129
00133         const int cropRight = 96;
00134
00138         const int cropTop = 96;
00139
00143         const int cropBottom = 96;
00144
00150         const int scaledWidth = 224;
00151
00157         const int scaledHeight = 224;
00158
00159     };
00160 }
```

8.89 FaceParsing.h File Reference

Provides a class implementing the face parsing pre-processing.

```
#include "Configuration.h"
#include "segmentations.h"
#include <ONNXRTSegmentation.h>
```

Classes

- class OFIQ_LIB::modules::segmentations::FaceParsing

    Class managing the separation of facial parts not occluded by non-facial parts from other parts.

Namespaces

- namespace cv

    OpenCV's namespace.
- namespace OFIQ_LIB

    Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::segmentations

    Provides segmentation-related implementations.

8.89.1 Detailed Description

Provides a class implementing the face parsing pre-processing.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.90 FaceParsing.h

Go to the documentation of this file.
```
00001
00028 #pragma once
00029
00030 #include "Configuration.h"
00031 #include "segmentations.h"
00032
00033 #include <ONNXRTSegmentation.h>
00034
00038 namespace cv
00039 {
00043     class Mat;
00044 }
00045
00049 namespace OFIQ_LIB::modules::segmentations
00050 {
00083     class FaceParsing : public SegmentationExtractorInterface
00084     {
00085     public:
```

```
00092        explicit FaceParsing(const Configuration& config);
00093
00097        ~FaceParsing() override = default;
00098
00099
00100    protected:
00126        OFIQ::Image UpdateMask(
00127            OFIQ_LIB::Session& session, modules::segmentations::SegmentClassLabels faceSegment)
     override;
00128
00129    private:
00130
00134        ONNXRuntimeSegmentation m_onnxRuntimeEnv;
00135
00141        std::shared_ptr<cv::Mat> segmentationImage;
00142
00148        const std::string modelConfigItem = "params.measures.FaceParsing.model_path";
00149
00153        const int imageSize = 400;
00154
00158        const int cropLeft = 30;
00159
00163        const int cropRight = 30;
00164
00168        const int cropTop = 0;
00169
00173        const int cropBottom = 60;
00174
00183        static cv::Mat CreateBlob(const cv::Mat& image, int i_imageSize_one_dim);
00184
00196        static std::shared_ptr<cv::Mat> CalculateClassIds(
00197            const cv::Mat& resultImage,
00198            int i_imageSize_one_dim);
00199
00200        /*/
00201         * @brief Derives the private member \link segmentationImage\endlink
00202         * from the facial image data provided by the session object.
00203         * @details Implements CNN processing step of \link
     OFIQ_LIB::modules::segmentations::FaceParsing::UpdateMask()
00204         * UpdateMask()\endlink.
00205         * @param session Session object containing the original facial image and pre-processing
     results
00206         * computed by the \link OFIQ_LIB::OFIQImpl::performPreprocessing()
00207         * OFIQImpl::performPreprocessing()\endlink method.
00208         */
00209        void SetImage(OFIQ_LIB::Session& session);
00210    };
00211 }
```

8.91 ONNXRTSegmentation.h File Reference

Helper class to manage the ONNXRuntime session object.

```
#include <vector>
#include <opencv2/opencv.hpp>
#include <onnxruntime_cxx_api.h>
```

Classes

- class ONNXRuntimeSegmentation

  Helper class to manage the ONNXRuntime session object.

8.91.1 Detailed Description

Helper class to manage the ONNXRuntime session object.

Author

OFIQ development team

8.92 ONNXRTSegmentation.h

Go to the documentation of this file.
```
00001
00027 #pragma once
00028
00029 #include <vector>
00030
00031 #include <opencv2/opencv.hpp>
00032 #include <onnxruntime_cxx_api.h>
00033
00039 class ONNXRuntimeSegmentation
00040 {
00041 private:
00042
00047     Ort::Env m_ortenv;
00048
00053     Ort::MemoryInfo m_memory_info = Ort::MemoryInfo::CreateCpu(OrtDeviceAllocator, OrtMemTypeCPU);
00054
00059     std::array<int64_t, 4> m_inputShape;
00060
00065     std::unique_ptr<Ort::Session> m_ort_session;
00066
00074     void init_session(const std::vector<uint8_t>& i_model_data, int64_t i_imageWidth, int64_t
    i_imageHeight);
00075
00076
00077 public:
00082     ONNXRuntimeSegmentation() = default;
00083
00088     ~ONNXRuntimeSegmentation() = default;
00089
00097     void initialize(
00098         const std::vector<uint8_t>& i_modelData, int64_t i_imageWidth, int64_t i_imageHeight);
00099
00105     size_t getNumberOfOutputNodes();
00106
00113     std::vector<Ort::Value> run( std::vector<float>&  i_netInput);
00114
00115 };
```

8.93 segmentations.h File Reference

Base class for the different implementation of segmentation algorithms.
```
#include "ofiq_lib.h"
#include "Session.h"
```

Classes

- class OFIQ_LIB::SegmentationExtractorInterface

    Base class for the different implementation of segmentation algorithms.

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.
- namespace OFIQ_LIB::modules
- namespace OFIQ_LIB::modules::segmentations

    Provides segmentation-related implementations.

Enumerations

- enum class OFIQ_LIB::modules::segmentations::SegmentClassLabels {
  OFIQ_LIB::modules::segmentations::background , OFIQ_LIB::modules::segmentations::skin , OFIQ_LIB::modules::segmentati
  , OFIQ_LIB::modules::segmentations::r_brow ,
  OFIQ_LIB::modules::segmentations::l_eye , OFIQ_LIB::modules::segmentations::r_eye , OFIQ_LIB::modules::segmentations::
  , OFIQ_LIB::modules::segmentations::l_ear ,
  OFIQ_LIB::modules::segmentations::r_ear , OFIQ_LIB::modules::segmentations::ear_r , OFIQ_LIB::modules::segmentations::
  , OFIQ_LIB::modules::segmentations::mouth ,
  OFIQ_LIB::modules::segmentations::u_lip , OFIQ_LIB::modules::segmentations::l_lip , OFIQ_LIB::modules::segmentations::ne
  , OFIQ_LIB::modules::segmentations::neck_l ,
  OFIQ_LIB::modules::segmentations::cloth , OFIQ_LIB::modules::segmentations::hair , OFIQ_LIB::modules::segmentations::ha
  , OFIQ_LIB::modules::segmentations::face }

    Enum class of the different face regioons that can be segmented.

8.93.1 Detailed Description

Base class for the different implementation of segmentation algorithms.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.94 segmentations.h

Go to the documentation of this file.
```
00001
00027 #pragma once
00028
00029 #include "ofiq_lib.h"
00030 #include "Session.h"
00031
00035 namespace OFIQ_LIB
00036 {
00040     namespace modules::segmentations
00041     {
00045         enum class SegmentClassLabels
00046         {
00050             background,
00054             skin,
00058             l_brow,
00062             r_brow,
00066             l_eye,
00070             r_eye,
00074             eye_g,
00078             l_ear,
00082             r_ear,
00086             ear_r,
00090             nose,
00094             mouth,
00098             u_lip,
00102             l_lip,
00106             neck,
00110             neck_l,
00114             cloth,
00118             hair,
00122             hat,
00126             face
00127         };
00128     }
00129
00137     class SegmentationExtractorInterface
00138     {
00139     public:
00144         virtual ~SegmentationExtractorInterface() = default;
00145
00153         OFIQ::Image& GetMask(
00154             OFIQ_LIB::Session& session, modules::segmentations::SegmentClassLabels faceSegment);
00155
00156     protected:
00157
00165         virtual OFIQ::Image UpdateMask(
00166             OFIQ_LIB::Session& session,
00167             modules::segmentations::SegmentClassLabels faceSegment) = 0;
00168
00173         std::string GetLastSessionId() const { return lastSessionId; };
00174
00175     private:
00180         std::string lastSessionId;
00185         std::map<modules::segmentations::SegmentClassLabels, OFIQ::Image> masks;
00186     };
00187 }
```

8.95 Configuration.h File Reference

Provides a configuration class for handling configurations.

```
#include <map>
#include <string>
#include <filesystem>
#include <tao/json/forward.hpp>
#include <tao/json/value.hpp>
```

Classes

- class OFIQ_LIB::Configuration

  Configuration class.

Namespaces

- namespace OFIQ_LIB

  Namespace for OFIQ implementations.

8.95.1 Detailed Description

Provides a configuration class for handling configurations.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.96 Configuration.h

Go to the documentation of this file.
```
00001
00029 #pragma once
00030
00031 #include <map>
00032 #include <string>
00033 #include <filesystem>
00034
00035 #include <tao/json/forward.hpp>
00036 #include <tao/json/value.hpp>
00037
00041 namespace OFIQ_LIB
00042 {
00049     class Configuration
00050     {
00051     public:
00058         Configuration(const std::string& configDir, const std::string& configFilename);
00059
00068         bool GetBool(const std::string& key, bool& value) const;
00069
00078         bool GetString(const std::string& key, std::string& value) const;
00079
00089         bool GetNumber(const std::string& key, double& value) const;
00090
00102         bool GetStringList(const std::string& key, std::vector<std::string>& value) const;
00103
00110         bool GetBool(const std::string& key) const;
00111
```

```
00118        std::string GetString(const std::string& key) const;
00119
00126        double GetNumber(const std::string& key) const;
00127
00135        std::string getDataDir() const;
00136
00143        void SetDataDir(std::string dataDir);
00144
00145    private:
00149        std::map<std::string, tao::json::value, std::less<» parameters;
00150
00156        std::filesystem::path m_dataDir;
00157    };
00158 }
```

8.97 image_io.h File Reference

Provides helper functions for reading/writing images from/to disk.

```
#include "ofiq_lib.h"
```

Namespaces

- namespace OFIQ_LIB

  Namespace for OFIQ implementations.

Functions

- OFIQ_EXPORT OFIQ::ReturnStatus OFIQ_LIB::readImage (const std::string &filename, OFIQ::Image &image)

  Read image from disk.

8.97.1 Detailed Description

Provides helper functions for reading/writing images from/to disk.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Author

OFIQ development team

8.98 image_io.h

```
00001
00027 #ifndef OFIQ_LIB_IMAGE_IO_H
00028 #define OFIQ_LIB_IMAGE_IO_H
00029
00030 #include "ofiq_lib.h"
00031
00035 namespace OFIQ_LIB {
00036
00044     OFIQ_EXPORT OFIQ::ReturnStatus
00045         readImage(const std::string& filename, OFIQ::Image& image);
00046 }
00047
00048 #endif
```

8.99 image_utils.h File Reference

Provides image utility functions such as color conversion, luminance computation etc.

```
#include "ofiq_lib.h"
#include "Session.h"
#include <opencv2/imgcodecs.hpp>
#include <opencv2/imgproc.hpp>
```

Namespaces

- namespace OFIQ_LIB

  Namespace for OFIQ implementations.

Typedefs

- using OFIQ_LIB::ExposureRange = std::array<int, 2>

Functions

- OFIQ_EXPORT double OFIQ_LIB::ColorConvert (double v)

  Converts a color as specified in ISO/IEC 29794-5.
- OFIQ_EXPORT double OFIQ_LIB::Cubic (double x, double k, double eps)

  Cubic flattening function.
- OFIQ_EXPORT void OFIQ_LIB::ConvertBGRToCIELAB (const cv::Mat &bgrImage, double &a, double &b)

  Computes CIELAB values a^* and b^* from a BGR image.
- OFIQ_EXPORT cv::Mat OFIQ_LIB::GetLuminanceImageFromBGR (const cv::Mat &bgrImage)

  Converts a BGR image to the luminance image.
- OFIQ_EXPORT void OFIQ_LIB::CalculateReferencePoints (const OFIQ::FaceLandmarks &landmarks, OFIQ::LandmarkPoint &leftEyeCenter, OFIQ::LandmarkPoint &rightEyeCenter, double &interEyeDistance, double &eyeMouthDistance)

  Computes the left eye center, the right eye center, the (planar) inter-eye-distance and the eye to mouth distance from facial landmarks.
- OFIQ_EXPORT void OFIQ_LIB::CalculateRegionOfInterest (cv::Rect &leftRegionOfInterest, cv::Rect &rightRegionOfInterest, const OFIQ::LandmarkPoint &leftEyeCenter, const OFIQ::LandmarkPoint &right← EyeCenter, const double interEyeDistance, const double eyeMouthDistance)

Extracts regions being of interest for some measures (e.g. NaturalColour).

- OFIQ_EXPORT void OFIQ_LIB::GetNormalizedHistogram (const cv::Mat &luminanceImage, const cv::Mat &maskImage, cv::Mat1f &histogram)

    Computes the normalized histogram from a luminance image in 256 chunks.

- OFIQ_EXPORT double OFIQ_LIB::CalculateExposure (const Session &session, const ExposureRange &exposureRange)

    Helper function for some measures.

- OFIQ_EXPORT double OFIQ_LIB::ComputeBrightnessAspect (const cv::Mat &luminanceImage, const cv↩ ::Mat &maskImage, const ExposureRange &exposureRange)

    Helper function for some measures.

8.99.1 Detailed Description

Provides image utility functions such as color conversion, luminance computation etc.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.100 image_utils.h

Go to the documentation of this file.
```
00001
00027 #pragma once
00028
00029 #include "ofiq_lib.h"
00030 #include "Session.h"
00031 #include <opencv2/imgcodecs.hpp>
00032 #include <opencv2/imgproc.hpp>
00033
00037 namespace OFIQ_LIB {
00038
00039     using ExposureRange = std::array<int, 2>;
00040
00047     OFIQ_EXPORT double ColorConvert(double v);
00048
00057     OFIQ_EXPORT double Cubic(double x, double k, double eps);
00058
00065     OFIQ_EXPORT void ConvertBGRToCIELAB(const cv::Mat& bgrImage, double& a, double& b);
```

```
00066
00074     OFIQ_EXPORT cv::Mat GetLuminanceImageFromBGR(const cv::Mat& bgrImage );
00075
00085     OFIQ_EXPORT void CalculateReferencePoints(const OFIQ::FaceLandmarks& landmarks,
00086         OFIQ::LandmarkPoint& leftEyeCenter,
00087         OFIQ::LandmarkPoint& rightEyeCenter,
00088         double& interEyeDistance,
00089         double& eyeMouthDistance);
00090
00104     OFIQ_EXPORT void CalculateRegionOfInterest(cv::Rect& leftRegionOfInterest,
00105         cv::Rect& rightRegionOfInterest,
00106         const OFIQ::LandmarkPoint& leftEyeCenter,
00107         const OFIQ::LandmarkPoint& rightEyeCenter,
00108         const double interEyeDistance, const double eyeMouthDistance);
00109
00117     OFIQ_EXPORT void GetNormalizedHistogram(const cv::Mat& luminanceImage, const cv::Mat& maskImage,
    cv::Mat1f& histogram);
00118
00132     OFIQ_EXPORT double CalculateExposure(const Session& session, const ExposureRange& exposureRange);
00133
00145     OFIQ_EXPORT double ComputeBrightnessAspect(
00146         const cv::Mat& luminanceImage, const cv::Mat& maskImage, const ExposureRange& exposureRange);
00147 }
```

8.101 NeuronalNetworkContainer.h File Reference

```
#include "detectors.h"
#include "landmarks.h"
#include "segmentations.h"
#include "poseEstimators.h"
```

Classes

- struct OFIQ_LIB::NeuronalNetworkContainer

  Neural network container for OFIQ's preprocessing steps.

Namespaces

- namespace OFIQ_LIB

  Namespace for OFIQ implementations.

8.102 NeuronalNetworkContainer.h

Go to the documentation of this file.
```
00001
00027 #pragma once
00028 #include "detectors.h"
00029 #include "landmarks.h"
00030 #include "segmentations.h"
00031 #include "poseEstimators.h"
00032
00036 namespace OFIQ_LIB
00037 {
00041     struct NeuronalNetworkContainer
00042     {
00055         NeuronalNetworkContainer(
00056             std::shared_ptr<FaceDetectorInterface> faceDetector,
00057             std::shared_ptr<FaceLandmarkExtractorInterface> landmarkExtractor,
00058             std::shared_ptr<SegmentationExtractorInterface> segmentationExtractor,
00059             std::shared_ptr<PoseEstimatorInterface> poseEstimator,
00060             std::shared_ptr<SegmentationExtractorInterface> faceOcclusionExtractor
00061             )
00062             : faceDetector{faceDetector},
```

```
00063            landmarkExtractor{landmarkExtractor},
00064            segmentationExtractor{segmentationExtractor},
00065            poseEstimator{poseEstimator},
00066            faceOcclusionExtractor{faceOcclusionExtractor}
00067        {
00068        }
00069
00073        std::shared_ptr<FaceDetectorInterface> faceDetector;
00074
00078        std::shared_ptr<FaceLandmarkExtractorInterface> landmarkExtractor;
00079
00085        std::shared_ptr<SegmentationExtractorInterface> segmentationExtractor;
00086
00092        std::shared_ptr<SegmentationExtractorInterface> faceOcclusionExtractor;
00093
00097        std::shared_ptr<PoseEstimatorInterface> poseEstimator;
00098    };
00099 }
```

8.103 OFIQError.h File Reference

Provides a class for the error handling within the QFIQ.

```
#include "ofiq_lib.h"
#include <string_view>
```

Classes

- class OFIQ_LIB::OFIQError

  Implementation of a custom exception.

Namespaces

- namespace OFIQ_LIB

  Namespace for OFIQ implementations.

8.103.1 Detailed Description

Provides a class for the error handling within the QFIQ.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.104 OFIQError.h

```
00001
00027 #pragma once
00028 #include "ofiq_lib.h"
00029 #include <string_view>
00030
00034 namespace OFIQ_LIB
00035 {
00040     class OFIQError : public std::exception
00041     {
00042     public:
00049         OFIQError(OFIQ::ReturnCode returnCode, std::string_view message);
00050
00056         const char* what() const noexcept override { return extendedMessage.c_str(); }
00057
00063         OFIQ::ReturnCode whatCode() const noexcept { return returnCode; }
00064
00065     private:
00070         OFIQ::ReturnCode returnCode;
00071
00076         std::string message;
00077
00082         std::string extendedMessage;
00083     };
00084 }
```

8.105 Session.h File Reference

The session class is the data container used to distribute the image and additional data, including the data computed during the pre-processing.

```
#include "ofiq_lib.h"
#include <opencv2/opencv.hpp>
```

Classes

- class OFIQ_LIB::Session

Namespaces

- namespace OFIQ_LIB

    Namespace for OFIQ implementations.

Typedefs

- using OFIQ_LIB::EulerAngle = std::array<double, 3>

8.105.1 Detailed Description

The session class is the data container used to distribute the image and additional data, including the data computed during the pre-processing.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Author

OFIQ development team

8.106 Session.h

Go to the documentation of this file.
```
00001
00028 #pragma once
00029
00030 #include "ofiq_lib.h"
00031 #include <opencv2/opencv.hpp>
00032
00036 namespace OFIQ_LIB
00037 {
00041     struct NeuronalNetworkContainer;
00042
00043     using EulerAngle = std::array<double, 3>;
00044
00051     class Session
00052     {
00053     public:
00054
00061         Session(const OFIQ::Image& image, OFIQ::FaceImageQualityAssessment& assessment)
00062             : _image{image},
00063               _assessment{assessment},
00064               id{GenerateId()}
00065         {
00066         }
00067
00072         const OFIQ::Image& image() const { return _image; }
00073
00078         OFIQ::FaceImageQualityAssessment& assessment() { return _assessment; }
00079
00085         const std::string& Id() const { return id; }
00086
00087         // use the session object as data container
00088
00094         void setDetectedFaces(const std::vector<OFIQ::BoundingBox>& i_boundingBoxes);
00095
00101         std::vector<OFIQ::BoundingBox> getDetectedFaces() const;
00102
```

```
00103 #ifdef OFIQ_SINGLE_FACE_PRESENT_WITH_TMETRIC
00104         void setLandmarksAllFaces(const std::vector<OFIQ::FaceLandmarks>& i_detectedLandmarks);
00105         std::vector<OFIQ::FaceLandmarks> getLandmarksAllFaces() const;
00106 #endif
00107
00113         void setPose(const EulerAngle& i_pose);
00114
00120         EulerAngle getPose() const;
00121
00127         void setLandmarks(const OFIQ::FaceLandmarks& i_landmarks);
00128
00134         OFIQ::FaceLandmarks getLandmarks() const;
00135
00136
00142         void setAlignedFaceLandmarks(const OFIQ::FaceLandmarks& i_landmarks);
00143
00149         OFIQ::FaceLandmarks getAlignedFaceLandmarks() const;
00150
00156         void setAlignedFaceTransformationMatrix(const cv::Mat & i_transformationMatrix);
00157
00158
00164         cv::Mat getAlignedFaceTransformationMatrix() const;
00165
00166
00172         void setAlignedFace(const cv::Mat & i_alignedFace);
00173
00179         cv::Mat getAlignedFace() const;
00180
00186         void setAlignedFaceLandmarkedRegion(const cv::Mat & i_alignedFaceRegion);
00187
00193         cv::Mat getAlignedFaceLandmarkedRegion() const;
00194
00200         void setFaceParsingImage(const cv::Mat& i_parsingImage);
00201
00207         cv::Mat getFaceParsingImage() const;
00208
00214         void setFaceOcclusionSegmentationImage(const cv::Mat& i_segmentationImage);
00215
00221         cv::Mat getFaceOcclusionSegmentationImage() const;
00222
00223     private:
00228         const OFIQ::Image& _image;
00229
00234         OFIQ::FaceImageQualityAssessment& _assessment;
00239         std::vector<OFIQ::BoundingBox> detectedFaces;
00240
00241 #ifdef OFIQ_SINGLE_FACE_PRESENT_WITH_TMETRIC
00242         std::vector<OFIQ::FaceLandmarks> landmarksAllFaces;
00243 #endif
00244
00249         EulerAngle pose;
00250
00255         OFIQ::FaceLandmarks landmarks;
00256
00261         OFIQ::FaceLandmarks alignedFaceLandmarks;
00262
00267         cv::Mat alignedFaceTransformationMatrix;
00268
00273         cv::Mat alignedFace;
00274
00279         cv::Mat alignedFacelandmarkedRegion;
00280
00285         cv::Mat faceParsingImage;
00286
00291         cv::Mat faceOcclusionSegmentationImage;
00292
00298         std::string GenerateId() const;
00299
00304         std::string id;
00305     };
00306 }
```

8.107   utils.h File Reference

Helper functions used by several classes.

```
#include "ofiq_lib.h"
```

Classes

- struct [Point2f](#)

    Representation of a point with floating point arithmetics.
- struct [OFIQ_LIB::Point2i](#)

    Representation of a point with integer arithmetics.

Namespaces

- namespace [cv](#)

    OpenCV's namespace.
- namespace [OFIQ_LIB](#)

    Namespace for OFIQ implementations.

Functions

- [OFIQ_EXPORT](#) void [OFIQ_LIB::makeSquareBoundingBoxWithPadding](#) (const [OFIQ::BoundingBox](#) &i_↩
bb, const cv::Mat &i_input_image, cv::Mat &o_output_image, [OFIQ::BoundingBox](#) &o_bb, [Point2i](#) &o_↩
translation_vector)

    Some computations, especially neural networks, need a squarred image as input. This funtion consumes a boundig box and an input image. The greater parameter of width or height is used to define the side length of the new squarred bounding box. The face will be centered in the bounding box. Padding is added if needed. The squarred bounding box is used generate a new cropped image, the o_output_image. Required translations are described by the translation vector o_translation_vector.
- [OFIQ_EXPORT](#) [OFIQ::BoundingBox](#) [OFIQ_LIB::makeSquareBoundingBox](#) (const [OFIQ::BoundingBox](#) &i↩
_bb)

    This function converts a non-squarred bounding box into an squarred one. The side length is defined by the greater one of height or width.
- [OFIQ_EXPORT](#) size_t [OFIQ_LIB::findLargestBoundingBox](#) (const std::vector< [OFIQ::BoundingBox](#) > &faceRects)

    This function returns the position of the largest bounding box (largest in terms of area) from a vector of bounding boxes.
- [OFIQ_EXPORT](#) cv::Mat [OFIQ_LIB::copyToCvImage](#) (const [OFIQ::Image](#) &sourceImage, bool asGray↩
Image=false)

    Convert images in OFIQ::Image format into the OpenCV cv::Mat format. The image can be converted from color to gray scale by setting the parameter asGrayImage to true.
- [OFIQ_EXPORT](#) cv::Mat [OFIQ_LIB::alignImage](#) (const [OFIQ::Image](#) &faceImage, const [OFIQ::FaceLandmarks](#) &faceLandmarks, [OFIQ::FaceLandmarks](#) &alignedFaceLandmarks, cv::Mat &transformationMatrix)

    This function transforms a face image so that the position of the eyes, nose and mouth are roughly at a pre-defined position. Face alignment is the translation, rotation and scaling of the image to do this.
- [OFIQ_EXPORT](#) void [OFIQ_LIB::calculateEyeCenter](#) (const [OFIQ::FaceLandmarks](#) &faceLandmarks, [Point2f](#) &leftEyeCenter, [Point2f](#) &rightEyeCenter)

    Based on face landmarks the center of the left and right eye are computed.
- [OFIQ_EXPORT](#) [OFIQ::Image](#) [OFIQ_LIB::MakeGreyImage](#) (uint16_t width, uint16_t height)

    This function generates a gray scaled image with the resolution passed by the call.
- [OFIQ_EXPORT](#) float [OFIQ_LIB::tmetric](#) (const [OFIQ::FaceLandmarks](#) &faceLandmarks)

    Based on the provided landmarks this function computes the distance between the point between the eyes and the chin.
- [OFIQ_EXPORT](#) void [OFIQ_LIB::rotationMatrixToEulerAngles](#) (const cv::Mat &R, std::vector< double > &angles)

    Based on a given rotation matrix this functions computes and returns the corresponding Euler angles.

8.107.1 Detailed Description

Helper functions used by several classes.

Copyright

Copyright (c) 2024 Federal Office for Information Security, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN-CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Author

OFIQ development team

8.108 utils.h

Go to the documentation of this file.
```
00001
00027 #ifndef OFIQ_LIB_UTILS_H
00028 #define OFIQ_LIB_UTILS_H
00029
00030 #include "ofiq_lib.h"
00031
00035 namespace cv
00036 {
00040     class Mat;
00041 }
00042
00047 struct Point2f
00048 {
00049     float x;
00050     float y;
00051 };
00052
00053
00054
00055 namespace OFIQ_LIB
00056 {
00061     struct Point2i
00062     {
00063         int x;
00064         int y;
00065     };
00066
00081     OFIQ_EXPORT void makeSquareBoundingBoxWithPadding(
00082         const OFIQ::BoundingBox& i_bb,
00083         const cv::Mat& i_input_image,
00084         cv::Mat& o_output_image,
00085         OFIQ::BoundingBox& o_bb,
00086         Point2i & o_translation_vector
00087         );
00088
00095     OFIQ_EXPORT OFIQ::BoundingBox makeSquareBoundingBox(
00096         const OFIQ::BoundingBox& i_bb);
```

```
00097
00104      OFIQ_EXPORT size_t findLargestBoundingBox(
00105          const std::vector<OFIQ::BoundingBox>& faceRects);
00106
00114      OFIQ_EXPORT cv::Mat copyToCvImage(const OFIQ::Image& sourceImage, bool asGrayImage = false);
00115
00125      OFIQ_EXPORT cv::Mat alignImage(
00126          const OFIQ::Image& faceImage,
00127          const OFIQ::FaceLandmarks& faceLandmarks,
00128          OFIQ::FaceLandmarks& alignedFaceLandmarks,
00129          cv::Mat& transformationMatrix);
00130
00138      OFIQ_EXPORT void calculateEyeCenter(
00139          const OFIQ::FaceLandmarks& faceLandmarks,
00140          Point2f& leftEyeCenter,
00141          Point2f& rightEyeCenter);
00142
00150      OFIQ_EXPORT OFIQ::Image MakeGreyImage(uint16_t width, uint16_t height);
00151
00158      OFIQ_EXPORT float tmetric(const OFIQ::FaceLandmarks& faceLandmarks);
00159
00166      OFIQ_EXPORT void rotationMatrixToEulerAngles(const cv::Mat& R, std::vector<double>& angles);
00167 }
00168
00169 #endif
```

Index