

# Open Source Face Image Quality (OFIQ)

Generated by Doxygen 1.10.0



<b>1 Open Source Face Image Quality (OFIQ) Library</b>	<b>1</b>
1.1 Introduction	1
1.2 License	1
1.2.1 OFIQ License	1
1.2.2 License of dependencies	1
1.3 Compilation	4
1.3.1 Linux	4
1.3.1.1 Ubuntu 22.04 (x86_64)	4
1.3.1.2 Ubuntu 24.04 (x86_64)	5
1.3.1.3 Ubuntu 24.04 (Raspberry Pi/ARMv8)	5
1.3.2 Windows	6
1.3.2.1 Windows (x86_64)	6
1.3.2.2 Windows (x86)	7
1.4 MacOS	7
1.4.0.1 MacOS (ARM64)	7
1.4.1 MacOS (x86_64)	8
1.4.2 Download model files	8
1.4.3 Download conformance test images	8
1.4.4 Download of external libraries	9
1.4.5 Building without conan	9
1.5 Running conformance tests	10
1.6 Running the sample executable	10
1.6.1 Quality assessment for a single facial image	10
1.6.2 Quality assessment for multiple images	10
1.6.3 Arguments	10
1.7 Supported platforms	11
1.8 Precompiled binaries	12
1.9 Configuration	12
1.9.1 Configuration of the face detector	13
1.9.2 Configuration of the landmark extractor	14
1.9.3 Other required configurations	14
1.9.4 Requesting measures	14
1.9.5 Default configuration	15
1.9.6 Configuration of the quality mapping	19
1.10 C++ API	20
1.11 Implementation and pre-processing workflow	21
1.12 Tutorial: Extending OFIQ	22
1.13 Release notes	25
1.13.1 Changelog	27
1.13.1.1 Version 1.0.2 (2025-04-10)	27
1.13.1.2 Version 1.0.1 (2025-03-11)	27
1.13.1.3 Version 1.0.0 (2025-03-07)	27

1.13.1.4 Version 1.0.0-RC.2 (2024-07-31)	28
1.13.1.5 Version 1.0.0-RC.1 (2024-03-15)	28
<b>2 Namespace Index</b>	<b>29</b>
2.1 Namespace List	29
<b>3 Hierarchical Index</b>	<b>31</b>
3.1 Class Hierarchy	31
<b>4 Class Index</b>	<b>33</b>
4.1 Class List	33
<b>5 File Index</b>	<b>37</b>
5.1 File List	37
<b>6 Namespace Documentation</b>	<b>41</b>
6.1 cv Namespace Reference	41
6.1.1 Detailed Description	41
6.2 OFIQ Namespace Reference	41
6.2.1 Detailed Description	42
6.2.2 Typedef Documentation	42
6.2.2.1 Landmarks	42
6.2.2.2 QualityAssessments	43
6.2.3 Enumeration Type Documentation	43
6.2.3.1 FaceDetectorType	43
6.2.3.2 LandmarkType	43
6.2.3.3 QualityMeasure	43
6.2.3.4 QualityMeasureReturnCode	44
6.2.3.5 ReturnCode	44
6.2.4 Function Documentation	45
6.2.4.1 operator<<()	45
6.3 OFIQ_LIB Namespace Reference	45
6.3.1 Detailed Description	47
6.3.2 Typedef Documentation	47
6.3.2.1 EulerAngle	47
6.3.2.2 ExposureRange	47
6.3.3 Function Documentation	47
6.3.3.1 alignImage()	47
6.3.3.2 CalculateExposure()	48
6.3.3.3 calculateEyeCenter()	48
6.3.3.4 CalculateReferencePoints()	49
6.3.3.5 CalculateRegionOfInterest()	49
6.3.3.6 ColorConvert()	49
6.3.3.7 ComputeBrightnessAspect()	50

6.3.3.8 ConvertBGRToCIELAB()	50
6.3.3.9 copyToCvImage()	51
6.3.3.10 Cubic()	51
6.3.3.11 findLargestBoundingBox()	51
6.3.3.12 GetLuminanceImageFromBGR()	52
6.3.3.13 GetNormalizedHistogram()	52
6.3.3.14 MakeGreyImage()	52
6.3.3.15 makeSquareBoundingBox()	53
6.3.3.16 makeSquareBoundingBoxWithPadding()	53
6.3.3.17 readImage()	53
6.3.3.18 readImageFromByteArray()	55
6.3.3.19 tmetric()	55
6.4 OFIQ_LIB::modules Namespace Reference	55
6.5 OFIQ_LIB::modules::detectors Namespace Reference	56
6.5.1 Detailed Description	56
6.6 OFIQ_LIB::modules::landmarks Namespace Reference	56
6.6.1 Detailed Description	57
6.6.2 Typedef Documentation	57
6.6.2.1 FaceMap	57
6.6.2.2 FacePairMap	57
6.6.2.3 LandmarkId	58
6.6.2.4 LandmarkIdPair	58
6.6.2.5 LandmarkIdPairs	58
6.6.2.6 LandmarkIds	58
6.6.3 Enumeration Type Documentation	58
6.6.3.1 FaceParts	58
6.7 OFIQ_LIB::modules::landmarks::adnet Namespace Reference	59
6.7.1 Detailed Description	59
6.7.2 Variable Documentation	59
6.7.2.1 chin	59
6.7.2.2 contour	60
6.7.2.3 FaceMap	60
6.7.2.4 FacePairMap	60
6.7.2.5 forehead	60
6.7.2.6 leftEye	60
6.7.2.7 leftEyeCorners	61
6.7.2.8 mouthInner	61
6.7.2.9 mouthOuter	61
6.7.2.10 nosetip	61
6.7.2.11 pairsInnerLip	61
6.7.2.12 pairsLeftEye	61
6.7.2.13 pairsMouthCenter	62

6.7.2.14 pairsRightEye . . . . .	62
6.7.2.15 rightEye . . . . .	62
6.7.2.16 rightEyeCorners . . . . .	62
6.8 OFIQ_LIB::modules::measures Namespace Reference . . . . .	62
6.8.1 Detailed Description . . . . .	64
6.8.2 Function Documentation . . . . .	64
6.8.2.1 log() . . . . .	64
6.8.3 Variable Documentation . . . . .	64
6.8.3.1 ExecutorLogActive . . . . .	64
6.9 OFIQ_LIB::modules::poseEstimators Namespace Reference . . . . .	64
6.9.1 Detailed Description . . . . .	65
6.10 OFIQ_LIB::modules::segmentations Namespace Reference . . . . .	65
6.10.1 Detailed Description . . . . .	65
6.10.2 Enumeration Type Documentation . . . . .	65
6.10.2.1 SegmentClassLabels . . . . .	65
<b>7 Class Documentation</b> . . . . .	<b>67</b>
7.1 OFIQ_LIB::modules::landmarks::ADNetFaceLandmarkExtractor Class Reference . . . . .	67
7.1.1 Detailed Description . . . . .	68
7.1.2 Constructor & Destructor Documentation . . . . .	68
7.1.2.1 ADNetFaceLandmarkExtractor() . . . . .	68
7.1.2.2 ~ADNetFaceLandmarkExtractor() . . . . .	68
7.1.3 Member Function Documentation . . . . .	68
7.1.3.1 updateLandmarks() . . . . .	68
7.1.4 Member Data Documentation . . . . .	69
7.1.4.1 landmarkExtractor_ . . . . .	69
7.2 OFIQ_LIB::modules::measures::BackgroundUniformity Class Reference . . . . .	69
7.2.1 Detailed Description . . . . .	71
7.2.2 Constructor & Destructor Documentation . . . . .	71
7.2.2.1 BackgroundUniformity() . . . . .	71
7.2.3 Member Function Documentation . . . . .	71
7.2.3.1 Execute() . . . . .	71
7.2.4 Member Data Documentation . . . . .	71
7.2.4.1 m_cropBottom . . . . .	71
7.2.4.2 m_cropLeft . . . . .	72
7.2.4.3 m_cropRight . . . . .	72
7.2.4.4 m_cropTop . . . . .	72
7.2.4.5 m_erosionKernelSize . . . . .	72
7.2.4.6 m_targetHeight . . . . .	72
7.2.4.7 m_targetWidth . . . . .	72
7.3 OFIQ::BoundingBox Struct Reference . . . . .	73
7.3.1 Detailed Description . . . . .	73

7.3.2 Constructor & Destructor Documentation	73
7.3.2.1 BoundingBox() [1/2]	73
7.3.2.2 BoundingBox() [2/2]	73
7.3.3 Member Data Documentation	74
7.3.3.1 faceDetector	74
7.3.3.2 height	74
7.3.3.3 width	74
7.3.3.4 xleft	74
7.3.3.5 ytop	74
7.4 OFIQ_LIB::modules::measures::CompressionArtifacts Class Reference	75
7.4.1 Detailed Description	76
7.4.2 Constructor & Destructor Documentation	76
7.4.2.1 CompressionArtifacts()	76
7.4.3 Member Function Documentation	77
7.4.3.1 Execute()	77
7.4.4 Member Data Documentation	77
7.4.4.1 m_crop	77
7.4.4.2 m_dim	77
7.4.4.3 m_onnxRuntimeEnv	78
7.5 OFIQ_LIB::Configuration Class Reference	78
7.5.1 Detailed Description	79
7.5.2 Constructor & Destructor Documentation	79
7.5.2.1 Configuration()	79
7.5.3 Member Function Documentation	79
7.5.3.1 GetBool() [1/2]	79
7.5.3.2 GetBool() [2/2]	79
7.5.3.3 getDataDir()	80
7.5.3.4 GetNumber() [1/2]	80
7.5.3.5 GetNumber() [2/2]	81
7.5.3.6 GetString() [1/2]	81
7.5.3.7 GetString() [2/2]	81
7.5.3.8 GetStringList()	82
7.5.3.9 setDataDir()	82
7.5.4 Member Data Documentation	82
7.5.4.1 m_dataDir	82
7.5.4.2 parameters	83
7.6 OFIQ_LIB::modules::measures::CropOfTheFacelImage Class Reference	83
7.6.1 Detailed Description	84
7.6.2 Constructor & Destructor Documentation	84
7.6.2.1 CropOfTheFacelImage()	84
7.6.3 Member Function Documentation	85
7.6.3.1 Execute()	85

7.7 OFIQ_LIB::modules::measures::DynamicRange Class Reference	85
7.7.1 Detailed Description	86
7.7.2 Constructor & Destructor Documentation	86
7.7.2.1 DynamicRange()	86
7.7.3 Member Function Documentation	87
7.7.3.1 Execute()	87
7.8 OFIQ_LIB::modules::measures::Executor Class Reference	87
7.8.1 Detailed Description	88
7.8.2 Constructor & Destructor Documentation	88
7.8.2.1 Executor()	88
7.8.3 Member Function Documentation	88
7.8.3.1 ExecuteAll()	88
7.8.3.2 GetMeasures()	88
7.8.4 Member Data Documentation	88
7.8.4.1 m_measures	88
7.9 OFIQ_LIB::modules::measures::ExpressionNeutrality Class Reference	89
7.9.1 Detailed Description	90
7.9.2 Constructor & Destructor Documentation	90
7.9.2.1 ExpressionNeutrality()	90
7.9.3 Member Function Documentation	91
7.9.3.1 Execute()	91
7.9.4 Member Data Documentation	91
7.9.4.1 m_classifier	91
7.9.4.2 m_onnxRuntimeEnvCNN1	91
7.9.4.3 m_onnxRuntimeEnvCNN2	91
7.10 OFIQ_LIB::modules::measures::EyesOpen Class Reference	92
7.10.1 Detailed Description	93
7.10.2 Constructor & Destructor Documentation	93
7.10.2.1 EyesOpen()	93
7.10.3 Member Function Documentation	93
7.10.3.1 Execute()	93
7.11 OFIQ_LIB::modules::measures::EyesVisible Class Reference	94
7.11.1 Detailed Description	95
7.11.2 Constructor & Destructor Documentation	95
7.11.2.1 EyesVisible()	95
7.11.3 Member Function Documentation	96
7.11.3.1 Execute()	96
7.12 OFIQ_LIB::FaceDetectorInterface Class Reference	96
7.12.1 Detailed Description	97
7.12.2 Constructor & Destructor Documentation	97
7.12.2.1 ~FaceDetectorInterface()	97
7.12.3 Member Function Documentation	97



7.12.3.1 detectFaces()	97
7.12.3.2 UpdateFaces()	97
7.13 OFIQ::FacelImageQualityAssessment Struct Reference	98
7.13.1 Detailed Description	98
7.13.2 Constructor & Destructor Documentation	98
7.13.2.1 FacelImageQualityAssessment() [1/2]	98
7.13.2.2 FacelImageQualityAssessment() [2/2]	98
7.13.3 Member Data Documentation	99
7.13.3.1 boundingBox	99
7.13.3.2 qAssessments	99
7.14 OFIQ_LIB::FaceLandmarkExtractorInterface Class Reference	99
7.14.1 Detailed Description	99
7.14.2 Constructor & Destructor Documentation	100
7.14.2.1 ~FaceLandmarkExtractorInterface()	100
7.14.3 Member Function Documentation	100
7.14.3.1 extractLandmarks()	100
7.14.3.2 updateLandmarks()	100
7.15 OFIQ::FaceLandmarks Struct Reference	101
7.15.1 Detailed Description	101
7.15.2 Constructor & Destructor Documentation	101
7.15.2.1 FaceLandmarks()	101
7.15.3 Member Data Documentation	101
7.15.3.1 landmarks	101
7.15.3.2 type	101
7.16 OFIQ_LIB::modules::landmarks::FaceMeasures Class Reference	102
7.16.1 Detailed Description	102
7.16.2 Constructor & Destructor Documentation	102
7.16.2.1 FaceMeasures()	102
7.16.3 Member Function Documentation	102
7.16.3.1 GetDistance() [1/2]	102
7.16.3.2 GetDistance() [2/2]	103
7.16.3.3 GetFaceMask()	103
7.16.3.4 GetMaxPairDistance()	104
7.16.3.5 GetMiddle() [1/3]	104
7.16.3.6 GetMiddle() [2/3]	104
7.16.3.7 GetMiddle() [3/3]	105
7.16.3.8 InterEyeDistance()	105
7.17 OFIQ_LIB::modules::measures::FaceOcclusionPrevention Class Reference	106
7.17.1 Detailed Description	107
7.17.2 Constructor & Destructor Documentation	107
7.17.2.1 FaceOcclusionPrevention()	107
7.17.3 Member Function Documentation	107

7.17.3.1 Execute()	107
7.18 OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation Class Reference	108
7.18.1 Detailed Description	109
7.18.2 Constructor & Destructor Documentation	109
7.18.2.1 FaceOcclusionSegmentation()	109
7.18.2.2 ~FaceOcclusionSegmentation()	110
7.18.3 Member Function Documentation	110
7.18.3.1 GetFaceOcclusionSegmentation()	110
7.18.3.2 UpdateMask()	110
7.18.4 Member Data Documentation	111
7.18.4.1 m_cropBottom	111
7.18.4.2 m_cropLeft	111
7.18.4.3 m_cropRight	111
7.18.4.4 m_cropTop	111
7.18.4.5 m_modelConfigItem	111
7.18.4.6 m_onnxRuntimeEnv	112
7.18.4.7 m_scaledHeight	112
7.18.4.8 m_scaledWidth	112
7.18.4.9 m_segmentationImage	112
7.19 OFIQ_LIB::modules::segmentations::FaceParsing Class Reference	112
7.19.1 Detailed Description	114
7.19.2 Constructor & Destructor Documentation	114
7.19.2.1 FaceParsing()	114
7.19.2.2 ~FaceParsing()	115
7.19.3 Member Function Documentation	115
7.19.3.1 CalculateClassIds()	115
7.19.3.2 CreateBlob()	115
7.19.3.3 SetImage()	116
7.19.3.4 UpdateMask()	116
7.19.4 Member Data Documentation	116
7.19.4.1 m_cropBottom	116
7.19.4.2 m_cropLeft	116
7.19.4.3 m_cropRight	117
7.19.4.4 m_imageSize	117
7.19.4.5 m_modelConfigItem	117
7.19.4.6 m_onnxRuntimeEnv	117
7.19.4.7 m_segmentationImage	117
7.20 OFIQ_LIB::modules::measures::HeadPose Class Reference	117
7.20.1 Detailed Description	119
7.20.2 Constructor & Destructor Documentation	119
7.20.2.1 HeadPose()	119
7.20.3 Member Function Documentation	119

7.20.3.1 Execute()	119
7.21 OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2 Class Reference	119
7.21.1 Detailed Description	121
7.21.2 Constructor & Destructor Documentation	121
7.21.2.1 HeadPose3DDFAV2()	121
7.21.2.2 ~HeadPose3DDFAV2()	121
7.21.3 Member Function Documentation	121
7.21.3.1 CropImage()	121
7.21.3.2 updatePose()	122
7.21.4 Member Data Documentation	122
7.21.4.1 m_expectedImageHeight	122
7.21.4.2 m_expectedImageNumberOfChannels	122
7.21.4.3 m_expectedImageWidth	122
7.21.4.4 m_inputShape	123
7.21.4.5 m_numberOfInputElements	123
7.21.4.6 m_ortenv	123
7.21.4.7 m_ortSession	123
7.21.4.8 m_paramPoseEstimatorModel	123
7.22 OFIQ_LIB::modules::measures::HeadSize Class Reference	123
7.22.1 Detailed Description	125
7.22.2 Constructor & Destructor Documentation	125
7.22.2.1 HeadSize()	125
7.22.3 Member Function Documentation	125
7.22.3.1 Execute()	125
7.23 OFIQ_LIB::modules::measures::IlluminationUniformity Class Reference	125
7.23.1 Detailed Description	127
7.23.2 Constructor & Destructor Documentation	127
7.23.2.1 IlluminationUniformity()	127
7.23.3 Member Function Documentation	127
7.23.3.1 Execute()	127
7.24 OFIQ::Image Struct Reference	128
7.24.1 Detailed Description	128
7.24.2 Constructor & Destructor Documentation	128
7.24.2.1 Image() [1/2]	128
7.24.2.2 Image() [2/2]	128
7.24.3 Member Function Documentation	129
7.24.3.1 deepcopy()	129
7.24.3.2 size()	129
7.24.4 Member Data Documentation	129
7.24.4.1 data	129
7.24.4.2 depth	130
7.24.4.3 height	130

7.24.4.4 width	130
7.25 OFIQ_LIB::modules::measures::InterEyeDistance Class Reference	130
7.25.1 Detailed Description	131
7.25.2 Constructor & Destructor Documentation	131
7.25.2.1 InterEyeDistance()	131
7.25.3 Member Function Documentation	132
7.25.3.1 Execute()	132
7.26 OFIQ::Interface Class Reference	132
7.26.1 Detailed Description	133
7.26.2 Constructor & Destructor Documentation	133
7.26.2.1 ~Interface()	133
7.26.3 Member Function Documentation	133
7.26.3.1 getImplementation()	133
7.26.3.2 getVersion()	133
7.26.3.3 initialize()	133
7.26.3.4 scalarQuality()	134
7.26.3.5 vectorQuality()	134
7.27 OFIQ_LIB::modules::landmarks::LandmarkPair Struct Reference	135
7.27.1 Detailed Description	135
7.27.2 Constructor & Destructor Documentation	135
7.27.2.1 LandmarkPair()	135
7.27.3 Member Data Documentation	136
7.27.3.1 Lower	136
7.27.3.2 Upper	136
7.28 OFIQ::LandmarkPoint Struct Reference	136
7.28.1 Detailed Description	137
7.28.2 Constructor & Destructor Documentation	137
7.28.2.1 LandmarkPoint() [1/2]	137
7.28.2.2 LandmarkPoint() [2/2]	137
7.28.3 Member Data Documentation	137
7.28.3.1 x	137
7.28.3.2 y	137
7.29 OFIQ_LIB::modules::measures::Luminance Class Reference	138
7.29.1 Detailed Description	139
7.29.2 Constructor & Destructor Documentation	139
7.29.2.1 Luminance()	139
7.29.3 Member Function Documentation	139
7.29.3.1 Execute()	139
7.30 OFIQ_LIB::modules::measures::Measure Class Reference	140
7.30.1 Detailed Description	141
7.30.2 Constructor & Destructor Documentation	141
7.30.2.1 Measure()	141

7.30.2.2 ~Measure()	142
7.30.3 Member Function Documentation	142
7.30.3.1 AddSigmoid() [1/2]	142
7.30.3.2 AddSigmoid() [2/2]	142
7.30.3.3 Execute()	143
7.30.3.4 ExecuteScalarConversion() [1/2]	143
7.30.3.5 ExecuteScalarConversion() [2/2]	143
7.30.3.6 ExpandKey()	144
7.30.3.7 GetMeasureName()	144
7.30.3.8 GetName()	144
7.30.3.9 GetQualityMeasure()	145
7.30.3.10 ScalarConversion()	145
7.30.3.11 SetQualityMeasure()	145
7.30.3.12 Sigmoid()	146
7.30.4 Member Data Documentation	146
7.30.4.1 configuration	146
7.30.4.2 m_measure	146
7.30.4.3 m_sigmoidMap	147
7.31 OFIQ_LIB::modules::measures::MeasureFactory Class Reference	147
7.31.1 Detailed Description	147
7.31.2 Constructor & Destructor Documentation	147
7.31.2.1 MeasureFactory()	147
7.31.3 Member Function Documentation	147
7.31.3.1 CreateMeasure()	147
7.32 OFIQ_LIB::modules::measures::MouthClosed Class Reference	148
7.32.1 Detailed Description	149
7.32.2 Constructor & Destructor Documentation	149
7.32.2.1 MouthClosed()	149
7.32.3 Member Function Documentation	150
7.32.3.1 Execute()	150
7.33 OFIQ_LIB::modules::measures::MouthOcclusionPrevention Class Reference	150
7.33.1 Detailed Description	151
7.33.2 Constructor & Destructor Documentation	151
7.33.2.1 MouthOcclusionPrevention()	151
7.33.3 Member Function Documentation	152
7.33.3.1 Execute()	152
7.34 OFIQ_LIB::modules::measures::NaturalColour Class Reference	152
7.34.1 Detailed Description	154
7.34.2 Constructor & Destructor Documentation	154
7.34.2.1 NaturalColour()	154
7.34.3 Member Function Documentation	154
7.34.3.1 CalculateScore()	154

7.34.3.2 CreateMaskedImage()	155
7.34.3.3 Execute()	155
7.34.3.4 ReducelImageToRegionOfInterest()	155
7.35 OFIQ_LIB::NeuronalNetworkContainer Struct Reference	156
7.35.1 Detailed Description	156
7.35.2 Constructor & Destructor Documentation	156
7.35.2.1 NeuronalNetworkContainer()	156
7.35.3 Member Data Documentation	157
7.35.3.1 faceDetector	157
7.35.3.2 faceOcclusionExtractor	157
7.35.3.3 landmarkExtractor	157
7.35.3.4 poseEstimator	157
7.35.3.5 segmentationExtractor	157
7.36 OFIQ_LIB::modules::measures::NoHeadCoverings Class Reference	158
7.36.1 Detailed Description	159
7.36.2 Constructor & Destructor Documentation	159
7.36.2.1 NoHeadCoverings()	159
7.36.3 Member Function Documentation	160
7.36.3.1 Execute()	160
7.36.4 Member Data Documentation	160
7.36.4.1 m_t0	160
7.36.4.2 m_t1	160
7.36.4.3 m_w	161
7.36.4.4 m_x0	161
7.37 OFIQ_LIB::OFIQError Class Reference	161
7.37.1 Detailed Description	162
7.37.2 Constructor & Destructor Documentation	162
7.37.2.1 OFIQError()	162
7.37.3 Member Function Documentation	162
7.37.3.1 what()	162
7.37.3.2 whatCode()	162
7.37.4 Member Data Documentation	163
7.37.4.1 m_extendedMessage	163
7.37.4.2 m_message	163
7.37.4.3 m_returnCode	163
7.38 OFIQ_LIB::OFIQImpl Class Reference	163
7.38.1 Detailed Description	164
7.38.2 Constructor & Destructor Documentation	164
7.38.2.1 OFIQImpl()	164
7.38.2.2 ~OFIQImpl()	165
7.38.3 Member Function Documentation	165
7.38.3.1 alignFacelImage()	165

7.38.3.2 CreateExecutor()	165
7.38.3.3 CreateNetworks()	165
7.38.3.4 initialize()	165
7.38.3.5 performPreprocessing()	166
7.38.3.6 scalarQuality()	166
7.38.3.7 vectorQuality()	166
7.38.4 Member Data Documentation	167
7.38.4.1 config	167
7.38.4.2 dummyAssement	167
7.38.4.3 dummyImage	167
7.38.4.4 m_emptySession	167
7.38.4.5 m_executorPtr	167
7.38.4.6 networks	168
7.39 ONNXRuntimeSegmentation Class Reference	168
7.39.1 Detailed Description	169
7.39.2 Constructor & Destructor Documentation	169
7.39.2.1 ONNXRuntimeSegmentation()	169
7.39.2.2 ~ONNXRuntimeSegmentation()	169
7.39.3 Member Function Documentation	169
7.39.3.1 getNumberOfOutputNodes()	169
7.39.3.2 init_session()	169
7.39.3.3 initialize()	170
7.39.3.4 run()	170
7.39.4 Member Data Documentation	170
7.39.4.1 m_inputShape	170
7.39.4.2 m_memoryInfo	170
7.39.4.3 m_ortenv	170
7.39.4.4 m_ortSession	171
7.40 OFIQ_LIB::modules::measures::OverExposurePrevention Class Reference	171
7.40.1 Detailed Description	172
7.40.2 Constructor & Destructor Documentation	172
7.40.2.1 OverExposurePrevention()	172
7.40.3 Member Function Documentation	173
7.40.3.1 Execute()	173
7.41 OFIQ_LIB::modules::landmarks::PartExtractor Class Reference	173
7.41.1 Detailed Description	173
7.41.2 Member Function Documentation	173
7.41.2.1 getFacePart()	173
7.41.2.2 getPairsForPart()	174
7.42 Point2f Struct Reference	174
7.42.1 Detailed Description	174
7.42.2 Member Data Documentation	175

7.42.2.1 x	175
7.42.2.2 y	175
7.43 OFIQ_LIB::Point2i Struct Reference	175
7.43.1 Detailed Description	175
7.43.2 Member Data Documentation	175
7.43.2.1 x	175
7.43.2.2 y	175
7.44 OFIQ_LIB::PoseEstimatorInterface Class Reference	176
7.44.1 Detailed Description	176
7.44.2 Member Typedef Documentation	176
7.44.2.1 EulerAngle	176
7.44.3 Constructor & Destructor Documentation	177
7.44.3.1 ~PoseEstimatorInterface()	177
7.44.4 Member Function Documentation	177
7.44.4.1 estimatePose()	177
7.44.4.2 updatePose()	177
7.44.5 Member Data Documentation	177
7.44.5.1 m_lastSessionId	177
7.44.5.2 m_pose	178
7.45 OFIQ::QualityMeasureResult Struct Reference	178
7.45.1 Detailed Description	178
7.45.2 Constructor & Destructor Documentation	178
7.45.2.1 QualityMeasureResult() [1/2]	178
7.45.2.2 QualityMeasureResult() [2/2]	178
7.45.3 Member Data Documentation	179
7.45.3.1 code	179
7.45.3.2 rawScore	179
7.45.3.3 scalar	179
7.46 OFIQ::ReturnStatus Struct Reference	179
7.46.1 Detailed Description	180
7.46.2 Constructor & Destructor Documentation	180
7.46.2.1 ReturnStatus() [1/2]	180
7.46.2.2 ReturnStatus() [2/2]	180
7.46.3 Member Data Documentation	180
7.46.3.1 code	180
7.46.3.2 info	181
7.47 OFIQ_LIB::SegmentationExtractorInterface Class Reference	181
7.47.1 Detailed Description	182
7.47.2 Constructor & Destructor Documentation	182
7.47.2.1 ~SegmentationExtractorInterface()	182
7.47.3 Member Function Documentation	182
7.47.3.1 GetLastSessionId()	182



7.47.3.2 GetMask()	182
7.47.3.3 UpdateMask()	183
7.47.4 Member Data Documentation	183
7.47.4.1 m_lastSessionId	183
7.47.4.2 m_masks	183
7.48 OFIQ_LIB::Session Class Reference	183
7.48.1 Detailed Description	185
7.48.2 Constructor & Destructor Documentation	185
7.48.2.1 Session()	185
7.48.3 Member Function Documentation	186
7.48.3.1 assessment()	186
7.48.3.2 GenerateId()	186
7.48.3.3 getAlignedFace()	186
7.48.3.4 getAlignedFaceLandmarkedRegion()	186
7.48.3.5 getAlignedFaceLandmarks()	187
7.48.3.6 getAlignedFaceTransformationMatrix()	187
7.48.3.7 getDetectedFaces()	187
7.48.3.8 getFaceOcclusionSegmentationImage()	187
7.48.3.9 getFaceParsingImage()	187
7.48.3.10 getLandmarks()	188
7.48.3.11 getPose()	188
7.48.3.12 Id()	188
7.48.3.13 image()	188
7.48.3.14 setAlignedFace()	188
7.48.3.15 setAlignedFaceLandmarkedRegion()	189
7.48.3.16 setAlignedFaceLandmarks()	189
7.48.3.17 setAlignedFaceTransformationMatrix()	189
7.48.3.18 setDetectedFaces()	189
7.48.3.19 setFaceOcclusionSegmentationImage()	190
7.48.3.20 setFaceParsingImage()	190
7.48.3.21 setLandmarks()	190
7.48.3.22 setPose()	190
7.48.4 Member Data Documentation	191
7.48.4.1 m_alignedFace	191
7.48.4.2 m_alignedFaceLandmarkedRegion	191
7.48.4.3 m_alignedFaceLandmarks	191
7.48.4.4 m_alignedFaceTransformationMatrix	191
7.48.4.5 m_assessment	191
7.48.4.6 m_detectedFaces	191
7.48.4.7 m_faceOcclusionSegmentationImage	192
7.48.4.8 m_faceParsingImage	192
7.48.4.9 m_id	192

7.48.4.10 m_image . . . . .	192
7.48.4.11 m_landmarks . . . . .	192
7.48.4.12 m_pose . . . . .	192
7.49 OFIQ_LIB::modules::measures::Sharpness Class Reference . . . . .	193
7.49.1 Detailed Description . . . . .	194
7.49.2 Constructor & Destructor Documentation . . . . .	194
7.49.2.1 Sharpness() . . . . .	194
7.49.3 Member Function Documentation . . . . .	195
7.49.3.1 Execute() . . . . .	195
7.49.3.2 GetClassifierFocusFeatures() . . . . .	195
7.49.3.3 GetCroppedImages() . . . . .	195
7.49.4 Member Data Documentation . . . . .	196
7.49.4.1 m_faceRegionAlpha . . . . .	196
7.49.4.2 m_modelFile . . . . .	196
7.49.4.3 m_numTrees . . . . .	196
7.49.4.4 m_rtree . . . . .	196
7.49.4.5 m_useAligned . . . . .	196
7.50 OFIQ_LIB::modules::measures::SigmoidParameters Struct Reference . . . . .	197
7.50.1 Detailed Description . . . . .	197
7.50.2 Constructor & Destructor Documentation . . . . .	198
7.50.2.1 SigmoidParameters() . . . . .	198
7.50.3 Member Function Documentation . . . . .	198
7.50.3.1 Reset() . . . . .	198
7.50.3.2 setInverse() . . . . .	198
7.50.4 Member Data Documentation . . . . .	198
7.50.4.1 a . . . . .	198
7.50.4.2 h . . . . .	198
7.50.4.3 round . . . . .	199
7.50.4.4 s . . . . .	199
7.50.4.5 w . . . . .	199
7.50.4.6 x0 . . . . .	199
7.51 OFIQ_LIB::modules::measures::SingleFacePresent Class Reference . . . . .	199
7.51.1 Detailed Description . . . . .	201
7.51.2 Constructor & Destructor Documentation . . . . .	201
7.51.2.1 SingleFacePresent() . . . . .	201
7.51.3 Member Function Documentation . . . . .	201
7.51.3.1 Execute() . . . . .	201
7.52 OFIQ_LIB::modules::detectors::SSDFaceDetector Class Reference . . . . .	201
7.52.1 Detailed Description . . . . .	202
7.52.2 Constructor & Destructor Documentation . . . . .	202
7.52.2.1 SSDFaceDetector() . . . . .	202
7.52.2.2 ~SSDFaceDetector() . . . . .	203

7.52.3 Member Function Documentation . . . . .	203
7.52.3.1 UpdateFaces() . . . . .	203
7.52.4 Member Data Documentation . . . . .	203
7.52.4.1 m_confidenceThreshold . . . . .	203
7.52.4.2 m_dnnNet . . . . .	203
7.52.4.3 m_minimalRelativeFaceSize . . . . .	204
7.52.4.4 m_padding . . . . .	204
7.53 OFIQ_LIB::modules::measures::UnderExposurePrevention Class Reference . . . . .	204
7.53.1 Detailed Description . . . . .	205
7.53.2 Constructor & Destructor Documentation . . . . .	205
7.53.2.1 UnderExposurePrevention() . . . . .	205
7.53.3 Member Function Documentation . . . . .	206
7.53.3.1 Execute() . . . . .	206
7.54 OFIQ_LIB::modules::measures::UnifiedQualityScore Class Reference . . . . .	206
7.54.1 Detailed Description . . . . .	207
7.54.2 Constructor & Destructor Documentation . . . . .	207
7.54.2.1 UnifiedQualityScore() . . . . .	207
7.54.3 Member Function Documentation . . . . .	208
7.54.3.1 Execute() . . . . .	208
7.54.4 Member Data Documentation . . . . .	208
7.54.4.1 m_onnxRuntimeEnv . . . . .	208
<b>8 File Documentation . . . . .</b>	<b>209</b>
8.1 mainpage.h File Reference . . . . .	209
8.1.1 Detailed Description . . . . .	209
8.2 mainpage.h . . . . .	209
8.3 ofiq_lib.h File Reference . . . . .	210
8.3.1 Detailed Description . . . . .	210
8.3.2 Macro Definition Documentation . . . . .	211
8.3.2.1 OFIQ_EXPORT . . . . .	211
8.4 ofiq_lib.h . . . . .	211
8.5 ofiq_lib_impl.h File Reference . . . . .	211
8.5.1 Detailed Description . . . . .	212
8.6 ofiq_lib_impl.h . . . . .	212
8.7 ofiq_structs.h File Reference . . . . .	213
8.7.1 Detailed Description . . . . .	215
8.8 ofiq_structs.h . . . . .	215
8.9 AllDetectors.h File Reference . . . . .	218
8.9.1 Detailed Description . . . . .	218
8.10 AllDetectors.h . . . . .	219
8.11 detectors.h File Reference . . . . .	219
8.11.1 Detailed Description . . . . .	219

8.12 detectors.h . . . . .	220
8.13 opencv_ssd_face_detector.h File Reference . . . . .	220
8.13.1 Detailed Description . . . . .	220
8.14 opencv_ssd_face_detector.h . . . . .	221
8.15 adnet_FaceMap.h File Reference . . . . .	221
8.15.1 Detailed Description . . . . .	223
8.16 adnet_FaceMap.h . . . . .	223
8.17 adnet_landmarks.h File Reference . . . . .	224
8.17.1 Detailed Description . . . . .	225
8.18 adnet_landmarks.h . . . . .	225
8.19 AllLandmarks.h File Reference . . . . .	225
8.19.1 Detailed Description . . . . .	226
8.20 AllLandmarks.h . . . . .	226
8.21 FaceMeasures.h File Reference . . . . .	226
8.21.1 Detailed Description . . . . .	227
8.22 FaceMeasures.h . . . . .	227
8.23 FaceParts.h File Reference . . . . .	228
8.23.1 Detailed Description . . . . .	229
8.24 FaceParts.h . . . . .	229
8.25 landmarks.h File Reference . . . . .	230
8.25.1 Detailed Description . . . . .	230
8.26 landmarks.h . . . . .	231
8.27 PartExtractor.h File Reference . . . . .	231
8.27.1 Detailed Description . . . . .	231
8.28 PartExtractor.h . . . . .	232
8.29 AllMeasures.h File Reference . . . . .	232
8.29.1 Detailed Description . . . . .	233
8.30 AllMeasures.h . . . . .	234
8.31 BackgroundUniformity.h File Reference . . . . .	234
8.31.1 Detailed Description . . . . .	234
8.32 BackgroundUniformity.h . . . . .	235
8.33 CompressionArtifacts.h File Reference . . . . .	235
8.33.1 Detailed Description . . . . .	236
8.34 CompressionArtifacts.h . . . . .	236
8.35 CropOfTheFacelImage.h File Reference . . . . .	237
8.35.1 Detailed Description . . . . .	237
8.36 CropOfTheFacelImage.h . . . . .	238
8.37 DynamicRange.h File Reference . . . . .	238
8.37.1 Detailed Description . . . . .	238
8.38 DynamicRange.h . . . . .	239
8.39 Executor.h File Reference . . . . .	239
8.39.1 Detailed Description . . . . .	240

8.40	Executor.h	241
8.41	ExpressionNeutrality.h File Reference	241
8.41.1	Detailed Description	242
8.42	ExpressionNeutrality.h	242
8.43	EyesOpen.h File Reference	242
8.43.1	Detailed Description	243
8.44	EyesOpen.h	243
8.45	EyesVisible.h File Reference	244
8.45.1	Detailed Description	244
8.46	EyesVisible.h	245
8.47	FaceOcclusionPrevention.h File Reference	245
8.47.1	Detailed Description	245
8.48	FaceOcclusionPrevention.h	246
8.49	HeadPose.h File Reference	246
8.49.1	Detailed Description	247
8.50	HeadPose.h	247
8.51	HeadSize.h File Reference	248
8.51.1	Detailed Description	248
8.52	HeadSize.h	249
8.53	IlluminationUniformity.h File Reference	249
8.53.1	Detailed Description	249
8.54	IlluminationUniformity.h	250
8.55	InterEyeDistance.h File Reference	250
8.55.1	Detailed Description	251
8.56	InterEyeDistance.h	251
8.57	Luminance.h File Reference	252
8.57.1	Detailed Description	252
8.58	Luminance.h	253
8.59	Measure.h File Reference	253
8.59.1	Detailed Description	253
8.60	Measure.h	254
8.61	MeasureFactory.h File Reference	255
8.61.1	Detailed Description	256
8.62	MeasureFactory.h	256
8.63	MouthClosed.h File Reference	257
8.63.1	Detailed Description	257
8.64	MouthClosed.h	258
8.65	MouthOcclusionPrevention.h File Reference	258
8.65.1	Detailed Description	258
8.66	MouthOcclusionPrevention.h	259
8.67	NaturalColour.h File Reference	259
8.67.1	Detailed Description	260

8.68 NaturalColour.h . . . . .	260
8.69 NoHeadCoverings.h File Reference . . . . .	261
8.69.1 Detailed Description . . . . .	261
8.70 NoHeadCoverings.h . . . . .	262
8.71 OverExposurePrevention.h File Reference . . . . .	262
8.71.1 Detailed Description . . . . .	262
8.72 OverExposurePrevention.h . . . . .	263
8.73 Sharpness.h File Reference . . . . .	263
8.73.1 Detailed Description . . . . .	264
8.74 Sharpness.h . . . . .	264
8.75 SingleFacePresent.h File Reference . . . . .	265
8.75.1 Detailed Description . . . . .	265
8.76 SingleFacePresent.h . . . . .	266
8.77 UnderExposurePrevention.h File Reference . . . . .	266
8.77.1 Detailed Description . . . . .	266
8.78 UnderExposurePrevention.h . . . . .	267
8.79 UnifiedQualityScore.h File Reference . . . . .	267
8.79.1 Detailed Description . . . . .	268
8.80 UnifiedQualityScore.h . . . . .	268
8.81 AllPoseEstimators.h File Reference . . . . .	269
8.81.1 Detailed Description . . . . .	269
8.82 AllPoseEstimators.h . . . . .	269
8.83 HeadPose3DDFAV2.h File Reference . . . . .	269
8.83.1 Detailed Description . . . . .	270
8.84 HeadPose3DDFAV2.h . . . . .	271
8.85 poseEstimators.h File Reference . . . . .	271
8.85.1 Detailed Description . . . . .	272
8.86 poseEstimators.h . . . . .	272
8.87 FaceOcclusionSegmentation.h File Reference . . . . .	273
8.87.1 Detailed Description . . . . .	273
8.88 FaceOcclusionSegmentation.h . . . . .	274
8.89 FaceParsing.h File Reference . . . . .	274
8.89.1 Detailed Description . . . . .	275
8.90 FaceParsing.h . . . . .	275
8.91 ONNXRTSegmentation.h File Reference . . . . .	276
8.91.1 Detailed Description . . . . .	276
8.92 ONNXRTSegmentation.h . . . . .	277
8.93 segmentations.h File Reference . . . . .	277
8.93.1 Detailed Description . . . . .	278
8.94 segmentations.h . . . . .	279
8.95 Configuration.h File Reference . . . . .	279
8.95.1 Detailed Description . . . . .	280

---

8.96 Configuration.h . . . . .	280
8.97 image_io.h File Reference . . . . .	281
8.97.1 Detailed Description . . . . .	281
8.98 image_io.h . . . . .	282
8.99 image_utils.h File Reference . . . . .	282
8.99.1 Detailed Description . . . . .	283
8.100 image_utils.h . . . . .	283
8.101 NeuronalNetworkContainer.h File Reference . . . . .	284
8.102 NeuronalNetworkContainer.h . . . . .	284
8.103 OFIQError.h File Reference . . . . .	285
8.103.1 Detailed Description . . . . .	285
8.104 OFIQError.h . . . . .	286
8.105 Session.h File Reference . . . . .	286
8.105.1 Detailed Description . . . . .	287
8.106 Session.h . . . . .	287
8.107 utils.h File Reference . . . . .	288
8.107.1 Detailed Description . . . . .	289
8.108 utils.h . . . . .	290
<b>Index</b>	<b>293</b>





# 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	<a href="https://cmake.org/">https://cmake.org/</a>	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.	<a href="https://conan.io/">https://conan.io/</a>	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-doxxygen.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.	<a href="https://www.doxxygen.org/">https://www.doxxygen.org/</a>	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.	<a href="https://www.doxxygen.org/">https://www.doxxygen.org/</a> , <a href="https://www.latex-project.org/">https://www.latex-project.org/</a>	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 conformance 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

#### 1.3.1.1 Ubuntu 22.04 (x86\_64)

Install necessary packages.

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

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 /path/to/OFIQ_Project/scripts
$ sh build.sh
```

where `/path/to/OFIQ_Project/` 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.1.17.3, and the header files in Release/include/.
data/models/	Model files downloaded from the ISO portal during build process.
data/tests/images/	Conformance test images downloaded from the ISO portal.

### 1.3.1.2 Ubuntu 24.04 (x86\_64)

Install necessary packages.

```
$ sudo apt-get install build-essential python3-pip cmake python3.12-venv
```

To install conan, a virtual Python environment needs to be generated first.

```
$ python3 -m venv /path/to/py_ofiq_env
```

where `/path/to/py_ofiq_env` is the path where the python environment will be stored and `py_ofiq_env` is the name of the new environment. Then install conan as follows.

```
$ source /path/to/py_ofiq_env/bin/activate
$ pip install conan==2.0.17
```

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

```
$ source /path/to/py_ofiq_env/bin/activate
$ cd /path/to/OFIQ_Project/scripts
$ sh build.sh
```

where `/path/to/OFIQ_Project/` 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 <code>OFIQSampleApp</code> and <code>libofiq_lib.so</code> .
build/conan/	Conan cache with packages downloaded.
install_x86_64_linux/	Folder with the installation including the binaries <code>Release/bin/OFIQSampleApp</code> , <code>Release/lib/libofiq_lib.so</code> , <code>Release/lib/libonnxruntime.so.1.17.3</code> , and the header files in <code>Release/include/</code> .
data/models/	Model files downloaded from the ISO portal during build process.
data/tests/images/	Conformance test images downloaded from the ISO portal.

### 1.3.1.3 Ubuntu 24.04 (Raspberry Pi/ARMv8)

The following has been tested on a Raspberry Pi5 with 4GB RAM.

Install necessary packages.

```
$ sudo apt-get install build-essential python3-pip cmake python3.12-venv
```

To install conan, a virtual Python environment needs to be generated first.

```
$ python3 -m venv /path/to/py_ofiq_env
```

where `/path/to/py_ofiq_env` is the path where the python environment will be stored and `py_ofiq_env` is the name of the new environment. Then install conan as follows.

```
$ source /path/to/py_ofiq_env/bin/activate
$ pip install conan==2.0.17
```

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

```
$ source /path/to/py_ofiq_env/bin/activate
$ cd /path/to/OFIQ_Project/scripts
$ sh build.sh --os linux-arm64
```

where `/path/to/OFIQ_Project/` 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_arm64_linux/	Folder with the installation including the binaries Release/bin/OFIQSampleApp, Release/lib/libofiq_lib.so, Release/lib/libonnxruntime.so.1.17.3, and the header files in Release/include/.
data/models/	Model files downloaded from the ISO portal during build process.
data/tests/images/	Conformance test images downloaded from the ISO portal.

## 1.3.2 Windows

### 1.3.2.1 Windows (x86\_64)

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 C:\Path\To\OFIQ_Project\scripts\
$ .\build.cmd
```

where `C:\Path\To\OFIQ_Project\` 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.1.17.3, and the header files in Release\include\.
data\models\	Model files downloaded from the ISO portal during build process.
data\tests\images\	Conformance test images downloaded from the ISO portal.

### 1.3.2.2 Windows (x86)

To compile Win32 binaries, one proceeds in the same way as for x86\_64 described above. But the building script is run with the `--arch x86` argument:

```
$ cd C:\Path\To\OFIQ_Project\scripts\  
$ .\build.cmd --arch x86
```

#

## 1.4 MacOS

### 1.4.0.1 MacOS (ARM64)

The following has been tested on macOS Sonoma Version 14.4.1 with ARM64 processor.

Install Homebrew

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Then update profiles

```
$ (echo; echo 'eval "$(/opt/homebrew/bin/brew shellenv)"') >> ~/.zprofile  
$ eval "$(/opt/homebrew/bin/brew shellenv)"
```

where `/opt/homebrew/bin/brew` is the path to the homebrew executable and can vary among MacOS systems.

NOTE The two commands from above for updating profiles are output by the Homebrew installation script.

Install CMake.

```
$ brew install cmake
```

Install Python using Miniconda by

```
$ mkdir -p /path/to/miniconda3/  
$ curl https://repo.anaconda.com/miniconda/Miniconda3-latest-MacOSX-arm64.sh -o /path/to/miniconda3/miniconda3.sh  
$ bash /path/to/miniconda3/miniconda.sh -b -u -p /path/to/miniconda3  
$ rm -rf /path/to/miniconda3/miniconda.sh  
$ /path/to/miniconda3/bin/conda init bash  
$ /path/to/miniconda3/bin/conda init zsh
```

where `/path/to/miniconda3` can be replaced by the path where Miniconda is installed.

Install conan.

```
$ python -m pip install conan==2.0.17
```

Finally, to build OFIQ run the following.

```
$ cd /path/to/OFIQ_Project/scripts/  
$ sh build.sh --os macos
```

#

### 1.4.1 MacOS (x86\_64)

To compile OFIQ on MacOS x86\_64 one needs to edit `/path/to/OFIQ_Project/conan/conan_profile_release_macos.txt` and `/path/to/OFIQ_Project/conan/conan_profile_debug_macos.txt` first. In both files replace the line

```
arch=armv8
```

by

```
arch=x86_64
```

Then apply the same actions as for MacOS compilation on ARM64.

### 1.4.2 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 so that the file structure matches the following.

<code>/path/to/OFIQ-Project/data/models/expression_neutrality/*</code>
<code>/path/to/OFIQ-Project/data/models/face_detection/*</code>
<code>/path/to/OFIQ-Project/data/models/face_landmark_estimation/*</code>
<code>/path/to/OFIQ-Project/data/models/face_occlusion_segmentation/*</code>
<code>/path/to/OFIQ-Project/data/models/face_parsing/*</code>
<code>/path/to/OFIQ-Project/data/models/head_pose_estimation/*</code>
<code>/path/to/OFIQ-Project/data/models/no_compression_artifacts/*</code>
<code>/path/to/OFIQ-Project/data/models/sharpness/*</code>
<code>/path/to/OFIQ-Project/data/models/unified_quality_score/*</code>

Here, `/path/to/OFIQ-Project/` denotes the path to OFIQ's root folder.

This step is integrated in the cmake building process.

### 1.4.3 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 so that the file structure matches the following.

<code>/path/to/OFIQ-Project/data/tests/images/b-01-smile.png</code>
<code>...</code>
<code>/path/to/OFIQ-Project/data/tests/images/r-09-background.png</code>

Here, `/path/to/OFIQ-Project/` denotes the path to OFIQ's root folder. Download and extraction of conformance test images can be performed when running the cmake building process.



### 1.4.4 Download of external libraries

The source of the external libraries that are used by OFIQ can be downloaded along with a full OFIQ release from [the ISO portal](#).

NOTE: At the date of publication, the link to the full OFIQ release were not available.

To build OFIQ without conan (described below), we need to extract the archive so that file structure matches the following:

/path/to/OFIQ-Project/extern/di/*
/path/to/OFIQ-Project/extern/flatbuffers/*
/path/to/OFIQ-Project/extern/googletest/*
/path/to/OFIQ-Project/extern/json/*
/path/to/OFIQ-Project/extern/magic_enum/*
/path/to/OFIQ-Project/extern/onnxruntime/*
/path/to/OFIQ-Project/extern/opencv-4.5.5/*
/path/to/OFIQ-Project/extern/PEGTL/*
/path/to/OFIQ-Project/extern/spdlog/*

Here, /path/to/OFIQ-Project/ denotes the path to OFIQ's root folder.

Note, download and extraction of the external libraries can be done automatically by the cmake building process described below.

### 1.4.5 Building without conan

It is important to note that building without conan requires manual extraction of model files and external library dependencies (and perhaps conformance test images) before running the building scripts. Further note, the files have to be placed manually in the right directory as described above. The missing files can be downloaded along with the release archive published at [the ISO portal](#).

To build OFIQ without conan, one can run

```
$ cd C:\Path\To\OFIQ-Project\scripts\  
$ .\build.cmd --no-conan
```

on Windows or

```
$ cd /path/to/OFIQ-Project/scripts/  
$ sh build.sh --no-conan
```

on Linux (both x86\_64 and ARMv8) and MacOS. This will download and extract the model files, conformance test images, and external libraries from the ISO portal as described above.

To suppress download of the dependencies (e.g., when one wants to compile without an internet connection), one can run

```
$ cd C:\Path\To\OFIQ-Project\scripts\  
$ .\build.cmd --no-conan --no-download
```

on Windows or

```
$ cd /path/to/OFIQ-Project/scripts/  
$ sh build.sh --no-conan --no-download
```

on Linux and MacOS.

## 1.5 Running conformance tests

The conformance tests are executed by going to `/path/to/OFIQ_Project/scripts/` and run `conformance_tests.cmd` (Windows). `conformance_tests.sh` (Linux/x86\_64). `conformance_tests.sh -os linux-arm64` (Linux/ARMv8) `conformance_tests.sh -os macos` (MacOS).

## 1.6 Running the sample executable

In this section, we describe how to run the sample application of OFIQ after compilation.

### 1.6.1 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 /path/to/OFIQ-Project/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 C:\Path\To\OFIQ-Project\install_x86_64\Release\bin
$ .\OFIQSampleApp -c ../../../../data/ofiq_config.jaxn -i
  ../../../../data/tests/images\b-01-smile.png
```

### 1.6.2 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 /path/to/OFIQ-Project/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 `/path/to/OFIQ_Project/install_x86_64_linux/Release/bin/table.csv`.

On Windows run the following commands.

```
$ cd C:\Path\To\OFIQ-Project\install_x86_64\Release\bin\
$ .\OFIQSampleApp -c ../../../../data\ -i
  ../../../../data\tests\images\ -o table.csv
```

The result will be written in the file `C:\Path\To\OFIQ-Project\install_x86_64\Release\bin\table.csv`.

### 1.6.3 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 <a href="#">doc/refman.pdf</a> ).
-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.

## 1.7 Supported platforms

OFIQ compilation has been successfully tested to compile and run on a variety of platforms. The following table gives an overview of platforms that have been successfully tested by the OFIQ development team.

platform	compiler	compiles via building script?	remarks
Ubuntu 22.04 (x86_64)	g++ 11.4.0	yes	-
Ubuntu 24.04 (x86_64)	g++ 13.2.0	yes	-
Ubuntu 24.04 (Raspberry Pi/ARM64)	g++ 13.3.0	yes	Building script needs specification of the argument <code>-os linux-arm64</code> .
Windows 10 (x86_64/win64)	Visual Studio 2019	yes	To compile external libraries, i.e., to run compilation without conan, an installation of Visual Studio 2022 was used which was required to build the onnxruntime dependency.
Windows 10 (x86/win32)	Visual Studio 2019	yes	Building script needs specification of the argument <code>-arch x86</code> . To compile external libraries, i.e., to run compilation without conan, an installation of Visual Studio 2022 was used which was required to build the onnxruntime dependency.
MacOS (ARM64)	clang 15.0.0	yes	Building script needs specification of the argument <code>-os macos</code> .
MacOS (x86_64)	clang 15.0.0	yes	Building script needs specification of the arguments <code>-os macos</code> . Furthermore, see the details on compiling for MacOS (x86_64) above.
Android	clang 12.0.8	no	Manual compilation including linking of dependencies is required. Not supported by OFIQ's building scripts.

iOS	clang 15.0.0	no	Manual compilation including linking of dependencies is required. Not supported by OFIQ's building scripts.
-----	--------------	----	---

OFIQ may compile on other configurations that have not been tested by the OFIQ development team. Compilation for 32-bit Linux systems, however, were not successful due to the onnxruntime being not supported by 32-bit gcc compiler. Also, compilation on Windows using gcc (for 32- nor 64-bit) did not work since onnxruntime did not build.

## 1.8 Precompiled binaries

A set of pre-compiled library binaries and the OFIQSampleApp will be made available on the ISO portal: <https://standards.iso.org/iso-iec/29794/-5/ed-1/en/>

## 1.9 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.

### 1.9.1 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 CAFFE format
prototxt_path	path to SSD's CAFFE prototype 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 $\text{min\_rel\_face\_size} * w$ 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.

### 1.9.2 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"
      }
    },
    ...
  }
}
```

### 1.9.3 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
<a href="#">HeadPose</a>	Head pose angles are pre-processed and used by some measures; therefore, the measure shall be configured. The path to the <a href="#">3DDFAV2</a> model file in ONNX format should be set using the <code>model_path</code> key.
<a href="#">FaceOcclusionSegmentation</a>	Face occlusion segmentation pre-processing used by some measures assessing occlusion prevention. The path to the <a href="#">FaceExtraction</a> model file in ONNX format should be set using the <code>model_path</code> 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.
<a href="#">FaceParsing</a>	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 <a href="#">BiSeNet</a> model file in ONNX format should be set using the <code>model_path</code> key.

### 1.9.4 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](#).

### 1.9.5 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-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 <a href="#">here</a>
-	Face detector	"config">"params". "detector"	-	see <a href="#">here</a>	-
-	Face landmark estimator	"config">"params". "landmarks"	-	see <a href="#">here</a>	-
-	Face parsing	"config". "params". "measures". "FaceParsing"	-	see <a href="#">here</a>	-
-	Face occlusion segmentation	"config". "params". "measures". "FaceOcclusionSegmentation"	-	see <a href="#">here</a>	-
-	Landmarked region	"config". "params". "measures". "FaceRegion"	-	alpha: is 0 per default and only used for internal purposes	-
0x41	Unified quality score	"config". "params". "measures". "UnifiedQualityScore"	"config". "measures". "UnifiedQualityScore"	model_path: Path to an iResNet50 model file in ONNX format	yes
0x42	Background uniformity	"config". "params". "measures". "BackgroundUniformity"	"config". "measures". "BackgroundUniformity"	none	yes

0x43	Illumination uniformity	"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	Absence of under-exposure	"config". "params". "measures". "Under↔ Exposure↔ Prevention"	"config". "measures". "Under↔ Exposure↔ Prevention"	none	yes
0x47	Absence 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	Absence of compression artifacts	"config". "params". "measures". "No↔ Compression↔ Artifacts"	"config". "measures". "No↔ Compression↔ Artifacts"	model_↔ path: Path to OFIQ's com- pression 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 openness	"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". "Leftward↔ CropOfThe↔ FacelImage"	"config". "measures". "CropOfThe↔ FacelImage"	none	yes
0x55	Rightward crop of the face image	"config">"params". "measures". "Rightward↔ CropOfThe↔ FacelImage"	"config". "measures". "CropOfThe↔ FacelImage"	none	yes
0x56	Margin above of the face image	"config">"params". "measures". "Margin↔ AboveOfThe↔ FacelImage"	"config". "measures". "CropOfThe↔ FacelImage"	none	yes
0x57	Margin below of the face image	"config">"params". "measures". "Margin↔ BelowOfThe↔ FacelImage"	"config". "measures". "CropOfThe↔ FacelImage"	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"	<p>cnn_model↵ _path1: Path to the CNN model <i>enet</i>↵ <i>b0_8_best</i>↵ <i>vgaf_embed</i>↵ <i>_zeroed.onnx</i> derived from <a href="#">here</a> in ONNX format</p> <p>cnn_model↵ _path2: Path to the CNN model <i>enet_b2_</i>↵ <i>8_embed</i>↵ <i>zeroed.</i>↵ <i>onnx</i> derived from <a href="#">here</a> in ONNX format.</p> <p>adaboost↵ _model_↵ path: Path to the AdaBoost classifier model file <i>hse</i>↵ <i>1_2_C</i>↵ <i>adaboost.</i>↵ <i>yml.gz</i> from <a href="#">here</a></p>	yes

0x5C	Absence of head coverings	"config">"params"."config". "measures". "NoHeadCovering"	"measures". "NoHeadCovering"	<p>T0 - Proportion of pixels classified as head covering <math>\leq</math> T0 will lead to a quality component value of 100 (best)</p> <p>T1 - Proportion of pixels classified as head covering <math>\geq</math> T1 will lead to a quality component value of 0 (worst)</p> <p>w - Proportion of pixels classified as head covering in (T0,T1) will be interpolated using a sigmoid function with w as standard deviation</p> <p>x0 - Proportion of pixels classified as head covering in (T0,T1) will be interpolated using a sigmoid function with x0 as development point</p>	no
------	---------------------------	--	---------------------------------	---	----

### 1.9.6 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 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 ( <code>std::round</code> ) 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
      }
    }
    ...
  }
}
```

## 1.10 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](#).

## 1.11 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::alignFacelImage\(\)](#).
  - (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.

## 1.12 Tutorial: Extending OFIQ

This section describes how to extend OFIQ by a new measure. We will choose an easy measure to illustrate the process. Therefore, we extend OFIQ by a somewhat dummy measure that assesses the subject's *non-surprisedness*. We will measure the surprisedness by the flatness of the eye-brow using the eye-brow landmarks.

Create a header file `NonSurprisedness.h` in the directory `/path/to/OFIQ-Project/OFIQlib/modules/measures/` with the following content

```
#pragma once

#include "landmarks.h"
#include "Measure.h"

namespace OFIQ_LIB::modules::measures
{
    class NonSurprisedness : public Measure
    {
    public:
        explicit NonSurprisedness(const Configuration& configuration);

        void Execute(OFIQ_LIB::Session & session) override;
    };
}
```

Also, create a source file `NonSurprisedness.cpp` in the directory `/path/to/OFIQ-Project/OFIQlib/modules/measures/src/` with the following content.

```
#include "NonSurprisedness.h"
#include "FaceMeasures.h"
#include "FaceParts.h"

using FaceMeasures = OFIQ_LIB::modules::landmarks::FaceMeasures;

namespace OFIQ_LIB::modules::measures
{
    static const auto qualityMeasure = OFIQ::QualityMeasure::NonSurprisedness;

    NonSurprisedness::NonSurprisedness(const Configuration& configuration)
        : Measure{ configuration, qualityMeasure }
    {
        // Implement me
    }

    void NonSurprisedness::Execute(OFIQ_LIB::Session & session)
    {
        // Implement me
    }
}
```

To make the source code to compile, we need to extend the enumeration `OFIQ::QualityMeasure` by the new measure. Therefore, edit the file `/path/to/OFIQ-Project/OFIQlib/include/ofiq_structs.h` and insert the item

```
NonSurprisedness
```

within the enumeration `OFIQ::QualityMeasure`. The resulting `ofiq_struct.h` could look like this.

```
...
namespace OFIQ
{
    ...
    enum class QualityMeasure
    {
        // UnifiedQualityScore
        UnifiedQualityScore = 0x41,
        ...
        // Newly inserted NonSurprisedness measure
        NonSurprisedness,
        // unknown measure
        NotSet = -1
    }
    ...
}
```

Then add the new header to `/path/to/OFIQ-Project/OFIQlib/modules/measures/AllMeasures.h`

```
...
#include "UnifiedQualityScore.h"
#include "NonSurprisedness.h"
```

and include the new measure in the file `/path/to/OFIQ-Project/OFIQlib/modules/measures/src/MeasureFactory.cpp`

```
static const std::map<OFIQ::QualityMeasure,
    std::function<std::unique_ptr<Measure>(const Configuration&)>> factoryMapping
{
    {OFIQ::QualityMeasure::SingleFacePresent, [] (const Configuration& configuration)
    { return std::make_unique<SingleFacePresent>(configuration); }},
    ...
    {OFIQ::QualityMeasure::NonSurprisedness, [] (const Configuration& configuration)
    { return std::make_unique<NonSurprisedness>(configuration); }},
}
```

If we are on Windows, we need to edit insert the new header file `NonSurprisedness.h` and the new source file `NonSurprisedness.cpp` in the file `/path/to/OFIQ-Project/OFIQLib/CMakeLists.windows.cmake`. The resulting `CMakeLists.windows.cmake` could look like this.

```
list(APPEND module_sources
    ${libImplementationSources}
    ...
    ${OFIQLIB_SOURCE_DIR}/modules/measures/src/NonSurprisedness.cpp
    ...

list(APPEND module_headers
    ${PUBLIC_HEADER_LIST}
    ...
    ${OFIQLIB_SOURCE_DIR}/modules/measures/NonSurprisedness.h
    ...
```

Likewise, if we are on Ubuntu or MacOS, we need to edit the files `CMakeLists.ubuntu.cmake` or `CMakeLists.macos.cmake`, respectively. Note, for Ubuntu or MacOS we only need to insert the file `NonSurprisedness.cpp` (and not the file `NonSurprisedness.h`).

After having finished the above steps, we may already build OFIQ by performing the steps described in the building steps. However, after compiling and running OFIQ, we may note that no output for our new no-surprisedness measure is made. Therefore, two additional steps need to be done:

1. Edit OFIQ's configuration file
2. Implement the member functions of the class `NonSurprisedness`

To edit OFIQ's default configuration file, open `/path/to/OFIQ-Project/data/ofiq_config.jaxn` and include the `NonSurprisedness` measure. The resulting `ofiq_config.jaxn` could look like this.

```
...
"config": {
  ...
  "measures": [
    ...
    "NonSurprisedness",
    ...
  ],
  ...
}
```

To implement the `Execute` method, we may want to compute the native quality measure first. At this point, we should define the non-surprisedness measure. For an eye-brow, we define the non-surprisedness using the difference of the y-coordinates of the northern eye-brow landmark to the y-coordinate between the left and the right eye-brow coordinate. To make the non-surprisedness independent from the facial image's resolution we divide the difference by the width of the eye-brow. the right eye-brow landmarks divided by the distance between the left and right The non-surprisedness is computed for the left and right eye-brow and then the minimum of the two eye brow's non-surprisedness as the overall non-surprisedness. The following functions computes the non-surprisedness for an eye-brow.

```
static double GetNonSurprisedness(
    const OFIQ::LandmarkPoint & north,
    const OFIQ::LandmarkPoint & left,
    const OFIQ::LandmarkPoint & right)
{
    auto middle = FaceMeasures::GetMiddle(OFIQ::Landmarks{ left, right });
    double diff = std::abs(middle.y-north.y);
    double width = FaceMeasures::GetDistance(left, right);
    return diff / width; // Results in NaN, when dividing by zero
}
```

The function `NonSurprisedness` uses functions provided by `FaceMeasures.h`. Now, we can implement the `Execute` method.

```
void NonSurprisedness::Execute(OFIQ_LIB::Session & session)
{
    auto landmarks = session.getAlignedFaceLandmarks();
    double leftNonSurprisedness = GetNonSurprisedness(landmarks.landmarks[33],
        landmarks.landmarks[38], landmarks.landmarks[35]);
    double rightNonSurprisedness = GetNonSurprisedness(landmarks.landmarks[50],
        landmarks.landmarks[46], landmarks.landmarks[44]);
    double nonSurprisedness =
```



```

        std::min(leftNonSurprisedness, rightNonSurprisedness);
    if (std::isnan(nonSurprisedness))
    {
        // If the measure is NaN, we cannot compute
        // the non-surprisedness and should indicate
        // this by 'FailureToAssess'
        SetQualityMeasure(session, qualityMeasure, nonSurprisedness,
            OFIQ::QualityMeasureReturnCode::FailureToAssess);
    }
    else
    {
        SetQualityMeasure(session, qualityMeasure, nonSurprisedness,
            OFIQ::QualityMeasureReturnCode::Success);
    }
}

```

The measure will automatically map the non-surprisedness measure to a quality values that is an integer between 0 and 100. For that, a default mapping is applied (see Section [Configuration of the quality mapping](#)). We should change the default mapping in the constructor. Therefore, we choose a mapping with the help of a sigmoid function that maps high non-surprisedness values to high quality values and vice-versa.

```

NonSurprisedness::NonSurprisedness(const Configuration& configuration)
    : Measure{ configuration, qualityMeasure }
{
    SigmoidParameters defaultValues;
    defaultValues.h = 100;
    defaultValues.x0 = 0.5;
    defaultValues.w = 1.0;
    defaultValues.round = true;
    AddSigmoid(qualityMeasure, defaultValues);
}

```

The above mapping is a fallback for the case when the mapping is not configured in the configuration file. To configure the mapping in the configuration file, edit the `ofiq_config.jaxn` so that it is of the form

```

"config": {
  ...
  "params" : {
    ...
    "measures": {
      ...
      "NonSurprisedness" : {
        "Sigmoid" : {
          "h": 100,
          "x0": 0.5,
          "w": 1.0,
          "round": true
        }
      }, ...
    }
  }
}

```

## 1.13 Release notes

This is OFIQ v1.0.2. 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.	<a href="#">UnifiedQualityScore</a>
0x42	Gradient-based background uniformity.	<a href="#">BackgroundUniformity</a>
0x43	Illumination uniformity by summing up the minima of the histograms of the left and the right side of the face.	<a href="#">IlluminationUniformity</a>
0x44	Luminance mean measure computed from the luminance histogram	<a href="#">Luminance</a>
0x45	Luminance variance measure computed from the luminance histogram	<a href="#">Luminance</a>
0x46	Under-exposure prevention by computing the proportion of low-intensity pixels in the luminance image to assess the absence of under-exposure	<a href="#">UnderExposurePrevention</a>
0x47	Over-exposure prevention by computing the proportion of high-intensity pixels in the luminance image to assess the absence of over-exposure	<a href="#">OverExposurePrevention</a>
0x48	Dynamic range computed from the luminance histogram.	<a href="#">DynamicRange</a>
0x49	Sharpness assessment based on a random forest classifier trained by the OFIQ development team.	<a href="#">Sharpness</a>
0x4A	Assessment of the absence of compression artifact (both JPEG and JPEG2000) based on a CNN trained by the OFIQ development team.	<a href="#">CompressionArtifacts</a>
0x4B	Assessment of the naturalness of the colour based on the conversion of the RGB presentation of the image to the CIELAB colour space.	<a href="#">NaturalColour</a>
0x4C	Assessment of the uniqueness of the most dominant face detected by comparing its size with the size of the second largest face detected	<a href="#">SingleFacePresent</a>
0x4D	Eyes openness assessment based on computing eyes aspect ratio from eye landmarks	<a href="#">EyesOpen</a>
0x4E	Mouth closed assessment based on computing a ratio from mouth landmarks	<a href="#">MouthClosed</a>
0x4F	Eyes visibility assessment by measuring the coverage of the eye visibility zone with the result of face occlusion segmentation computed during pre-processing.	<a href="#">EyesVisible</a>
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.	<a href="#">MouthOcclusionPrevention</a>
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.	<a href="#">FaceOcclusionPrevention</a>
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.	<a href="#">InterEyeDistance</a>
0x53	Size of the head based on computing the height of the face computed from facial landmarks with the height of the image.	<a href="#">HeadSize</a>
0x54	Leftward crop of the face image	<a href="#">CropOfTheFaceImage</a>

0x55	Rightward crop of the face image	<a href="#">CropOfTheFaceImage</a>
0x56	Margin above of the face image	<a href="#">CropOfTheFaceImage</a>
0x57	Margin below of the face image	<a href="#">CropOfTheFaceImage</a>
0x58	Pose angle yaw frontal alignment based on the <a href="#">3DDFAV2</a> .	<a href="#">HeadPose</a>
0x59	Pose angle pitch frontal alignment based on the <a href="#">3DDFAV2</a>	<a href="#">HeadPose</a>
0x5A	Pose angle roll frontal alignment based on the <a href="#">3DDFAV2</a>	<a href="#">HeadPose</a>
0x5B	Expression neutrality estimation based on a fusion of <a href="#">HSEMotion</a> with with <a href="#">Efficient-↔ Expression-Neutrality-Estimation</a> .	<a href="#">ExpressionNeutrality</a>
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.	<a href="#">NoHeadCoverings</a>

## 1.13.1 Changelog

### 1.13.1.1 Version 1.0.2 (2025-04-10)

- Supports compilation for Raspberry Pi/ARMv8 - see [https://github.com/BSI-OFIQ/OFIQ-↔ Project/pull/45](https://github.com/BSI-OFIQ/OFIQ-Project/pull/45)
- Fixes segmentation faults occurring on natural colour assessment in cases CIELAB values being attempted to be computed from an empty colour image
- EyesVisible component did not handle the cases where inter-eye distance is not computable. Now it does by resulting in FailureToAssess.
- IlluminationUniformity did not handle the case when histograms from empty subimages are to be computed. Now it does by resulting in FailureToAssess.
- FailureToAssess assessments of quality component do output a quality component value of -1 more consequently.

### 1.13.1.2 Version 1.0.1 (2025-03-11)

- OFIQ::Image struct had no method to import deep copies from binary data. This made in unpractical to use OFIQ using a binding such as Java/JNI. Now OFIQ::Images provides a method [deepcopy](#) for importing deep copies from binary data.  
This change resolves <https://github.com/BSI-OFIQ/OFIQ-Project/issues/63>.
- Code quality improvements as per Sonarqube checks (do not affect conformance tests)
- Memory management refinements
- Fixes memory allocations when cv::copyMakeBorder from OpenCV is invoked

### 1.13.1.3 Version 1.0.0 (2025-03-07)

First OFIQ release being the reference implementation for ISO/IEC 29794-5.

#### 1.13.1.4 Version 1.0.0-RC.2 (2024-07-31)

Second release of OFIQ's release candidate. The following changes have been implemented.

- Supports compilation on MacOS
- Successfully tested that it is possible (with some effort) to compile for and be conformant with mobile devices such as Android and iOS.
- Revisions as per up coming FDIS (e.g., quality mappings, update of conformance test table, etc.)
- Fix of the default config file: Changes to the CropOfTheFaceImage measures did not affect the quality mappings
- Fixes link to the OFIQ-MODELS.zip archive: <https://github.com/BSI-OFIQ/OFIQ-Project/issues/12>
- Removes libgtk dependency: <https://github.com/BSI-OFIQ/OFIQ-Project/issues/18>
- Removes Lapack from ubuntu cmake file which wasn't used: <https://github.com/BSI-OFIQ/OFIQ-Project/issues/20>
- Improves readability of source code by applying a style guide to member variables: <https://github.com/BSI-OFIQ/OFIQ-Project/issues/27>
- Fixes from static code analyses, for example:
  - <https://github.com/BSI-OFIQ/OFIQ-Project/issues/28>
  - <https://github.com/BSI-OFIQ/OFIQ-Project/issues/29>
  - <https://github.com/BSI-OFIQ/OFIQ-Project/issues/30>
  - <https://github.com/BSI-OFIQ/OFIQ-Project/issues/31>
  - and others
- Removes mis-leading usage hint in the sample app: <https://github.com/BSI-OFIQ/OFIQ-Project/issues/34>
- Avoids redundant RGB conversions: <https://github.com/BSI-OFIQ/OFIQ-Project/issues/36>
- Fixes a bug on continuous OpenCV matrices: <https://github.com/BSI-OFIQ/OFIQ-Project/issues/41>
- Fixes further issues and bugs and code beautification

#### 1.13.1.5 Version 1.0.0-RC.1 (2024-03-15)

Initial release of OFIQ's release candidate.

## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">cv</a>	OpenCV's namespace . . . . .	41
<a href="#">OFIQ</a>	Namespace for OFIQ API . . . . .	41
<a href="#">OFIQ_LIB</a>	Namespace for OFIQ implementations . . . . .	45
<a href="#">OFIQ_LIB::modules</a>	. . . . .	55
<a href="#">OFIQ_LIB::modules::detectors</a>	Provides face detector implementations . . . . .	56
<a href="#">OFIQ_LIB::modules::landmarks</a>	Provides implementations of a landmark extractors . . . . .	56
<a href="#">OFIQ_LIB::modules::landmarks::adnet</a>	Namespace for ADNet-specific landmarks . . . . .	59
<a href="#">OFIQ_LIB::modules::measures</a>	Provides measures implemented in OFIQ . . . . .	62
<a href="#">OFIQ_LIB::modules::poseEstimators</a>	Provides implementation of a head pose estimator . . . . .	64
<a href="#">OFIQ_LIB::modules::segmentations</a>	Provides segmentation-related implementations . . . . .	65



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

OFIQ::BoundingBox	73
OFIQ_LIB::Configuration	78
std::exception	
OFIQ_LIB::OFIQError	161
OFIQ_LIB::modules::measures::Executor	87
OFIQ_LIB::FaceDetectorInterface	96
OFIQ_LIB::modules::detectors::SSDFaceDetector	201
OFIQ::FaceImageQualityAssessment	98
OFIQ_LIB::FaceLandmarkExtractorInterface	99
OFIQ_LIB::modules::landmarks::ADNetFaceLandmarkExtractor	67
OFIQ::FaceLandmarks	101
OFIQ_LIB::modules::landmarks::FaceMeasures	102
OFIQ::Image	128
OFIQ::Interface	132
OFIQ_LIB::OFIQImpl	163
OFIQ_LIB::modules::landmarks::LandmarkPair	135
OFIQ::LandmarkPoint	136
OFIQ_LIB::modules::measures::Measure	140
OFIQ_LIB::modules::measures::BackgroundUniformity	69
OFIQ_LIB::modules::measures::CompressionArtifacts	75
OFIQ_LIB::modules::measures::CropOfTheFacelImage	83
OFIQ_LIB::modules::measures::DynamicRange	85
OFIQ_LIB::modules::measures::ExpressionNeutrality	89
OFIQ_LIB::modules::measures::EyesOpen	92
OFIQ_LIB::modules::measures::EyesVisible	94
OFIQ_LIB::modules::measures::FaceOcclusionPrevention	106
OFIQ_LIB::modules::measures::HeadPose	117
OFIQ_LIB::modules::measures::HeadSize	123
OFIQ_LIB::modules::measures::IlluminationUniformity	125
OFIQ_LIB::modules::measures::InterEyeDistance	130
OFIQ_LIB::modules::measures::Luminance	138
OFIQ_LIB::modules::measures::MouthClosed	148
OFIQ_LIB::modules::measures::MouthOcclusionPrevention	150
OFIQ_LIB::modules::measures::NaturalColour	152

OFIQ_LIB::modules::measures::NoHeadCoverings . . . . .	158
OFIQ_LIB::modules::measures::OverExposurePrevention . . . . .	171
OFIQ_LIB::modules::measures::Sharpness . . . . .	193
OFIQ_LIB::modules::measures::SingleFacePresent . . . . .	199
OFIQ_LIB::modules::measures::UnderExposurePrevention . . . . .	204
OFIQ_LIB::modules::measures::UnifiedQualityScore . . . . .	206
OFIQ_LIB::modules::measures::MeasureFactory . . . . .	147
OFIQ_LIB::NeuronalNetworkContainer . . . . .	156
ONNXRuntimeSegmentation . . . . .	168
OFIQ_LIB::modules::landmarks::PartExtractor . . . . .	173
Point2f . . . . .	174
OFIQ_LIB::Point2i . . . . .	175
OFIQ_LIB::PoseEstimatorInterface . . . . .	176
OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2 . . . . .	119
OFIQ::QualityMeasureResult . . . . .	178
OFIQ::ReturnStatus . . . . .	179
OFIQ_LIB::SegmentationExtractorInterface . . . . .	181
OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation . . . . .	108
OFIQ_LIB::modules::segmentations::FaceParsing . . . . .	112
OFIQ_LIB::Session . . . . .	183
OFIQ_LIB::modules::measures::SigmoidParameters . . . . .	197



## Chapter 4

# Class Index

### 4.1 Class List

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

<a href="#">OFIQ_LIB::modules::landmarks::ADNetFaceLandmarkExtractor</a>	
Class implementing the <a href="#">FaceLandmarkExtractorInterface</a> interface	67
<a href="#">OFIQ_LIB::modules::measures::BackgroundUniformity</a>	
Implementation of the background uniformity measure	69
<a href="#">OFIQ::BoundingBox</a>	
Data structure for describing bounding boxes, e.g. the face region of the faces found by a face detector	73
<a href="#">OFIQ_LIB::modules::measures::CompressionArtifacts</a>	
Implementation of the no compression artifacts measure	75
<a href="#">OFIQ_LIB::Configuration</a>	
Configuration class	78
<a href="#">OFIQ_LIB::modules::measures::CropOfTheFacelImage</a>	
Implementation of the crop of the face image measure	83
<a href="#">OFIQ_LIB::modules::measures::DynamicRange</a>	
Implementation of the dynamic range measure	85
<a href="#">OFIQ_LIB::modules::measures::Executor</a>	
This class takes care of the computation of the measures activated	87
<a href="#">OFIQ_LIB::modules::measures::ExpressionNeutrality</a>	
Provides a class implementing the expression neutrality measure	89
<a href="#">OFIQ_LIB::modules::measures::EyesOpen</a>	
Implementation of the eyes open measure	92
<a href="#">OFIQ_LIB::modules::measures::EyesVisible</a>	
Implementation of the eyes visible measure	94
<a href="#">OFIQ_LIB::FaceDetectorInterface</a>	
Provides the interface class to the face detector implementations	96
<a href="#">OFIQ::FacelImageQualityAssessment</a>	
Data structure storing the results of the different measurement computations	98
<a href="#">OFIQ_LIB::FaceLandmarkExtractorInterface</a>	
Implements the base class for the face landmark extractors	99
<a href="#">OFIQ::FaceLandmarks</a>	
Data structure for storing facial landmarks	101
<a href="#">OFIQ_LIB::modules::landmarks::FaceMeasures</a>	
Provides static functions doing computations with landmarks	102
<a href="#">OFIQ_LIB::modules::measures::FaceOcclusionPrevention</a>	
Implementation of the face occlusion prevention measure	106

<a href="#">OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation</a>	
Class managing the separation of facial parts not occluded by non-facial parts from other parts	108
<a href="#">OFIQ_LIB::modules::segmentations::FaceParsing</a>	
Class managing the separation of facial parts not occluded by non-facial parts from other parts	112
<a href="#">OFIQ_LIB::modules::measures::HeadPose</a>	
Implementation of head pose measures	117
<a href="#">OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2</a>	
Implementation of a head pose estimator	119
<a href="#">OFIQ_LIB::modules::measures::HeadSize</a>	
Implementation of the head size measure	123
<a href="#">OFIQ_LIB::modules::measures::IlluminationUniformity</a>	
Implementation of the illumination uniformity measure	125
<a href="#">OFIQ::Image</a>	
Struct representing a single image	128
<a href="#">OFIQ_LIB::modules::measures::InterEyeDistance</a>	
Implementation of the inter-eye distance measure	130
<a href="#">OFIQ::Interface</a>	
The interface to FACE QA implementation	132
<a href="#">OFIQ_LIB::modules::landmarks::LandmarkPair</a>	
Data container for storing pairs of landmarks	135
<a href="#">OFIQ::LandmarkPoint</a>	
Data structure to describe the x and y coordinate of a landmark	136
<a href="#">OFIQ_LIB::modules::measures::Luminance</a>	
Implementation of two luminance measures	138
<a href="#">OFIQ_LIB::modules::measures::Measure</a>	
Base class for measures implemented in OFIQ	140
<a href="#">OFIQ_LIB::modules::measures::MeasureFactory</a>	
Measure factor class	147
<a href="#">OFIQ_LIB::modules::measures::MouthClosed</a>	
Implementation of the mouth closed measure	148
<a href="#">OFIQ_LIB::modules::measures::MouthOcclusionPrevention</a>	
Implementation of the mouth occlusion prevention measure	150
<a href="#">OFIQ_LIB::modules::measures::NaturalColour</a>	
Implementation of the natural colour measure	152
<a href="#">OFIQ_LIB::NeuralNetworkContainer</a>	
Neural network container for OFIQ's preprocessing steps	156
<a href="#">OFIQ_LIB::modules::measures::NoHeadCoverings</a>	
Implementation of the no head covering measure	158
<a href="#">OFIQ_LIB::OFIQError</a>	
Implementation of a custom exception	161
<a href="#">OFIQ_LIB::OFIQImpl</a>	
Implementation of the OFIQ_LIB	163
<a href="#">ONNXRuntimeSegmentation</a>	
Helper class to manage the ONNXRuntime session object	168
<a href="#">OFIQ_LIB::modules::measures::OverExposurePrevention</a>	
Implementation of the over-exposure prevention measure	171
<a href="#">OFIQ_LIB::modules::landmarks::PartExtractor</a>	
Class that provides helper methods for the administration of landmarks	173
<a href="#">Point2f</a>	
Representation of a point with floating point arithmetics	174
<a href="#">OFIQ_LIB::Point2i</a>	
Representation of a point with integer arithmetics	175
<a href="#">OFIQ_LIB::PoseEstimatorInterface</a>	
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	176
<a href="#">OFIQ::QualityMeasureResult</a>	
Data structure to handle the results of a quality measure	178

<a href="#">OFIQ::ReturnStatus</a>	
A structure to contain information about a failure by the software under test . . . . .	179
<a href="#">OFIQ_LIB::SegmentationExtractorInterface</a>	
Base class for the different implementation of segmentation algorithms . . . . .	181
<a href="#">OFIQ_LIB::Session</a> . . . . .	183
<a href="#">OFIQ_LIB::modules::measures::Sharpness</a>	
Implementantation of the sharpness measure . . . . .	193
<a href="#">OFIQ_LIB::modules::measures::SigmoidParameters</a>	
Parameters of the sigmoid function based quality mapping . . . . .	197
<a href="#">OFIQ_LIB::modules::measures::SingleFacePresent</a>	
Implementation of the single face present measure . . . . .	199
<a href="#">OFIQ_LIB::modules::detectors::SSDFaceDetector</a>	
Implementation of a face detector using the SSD face detector CNN . . . . .	201
<a href="#">OFIQ_LIB::modules::measures::UnderExposurePrevention</a>	
Implementation of the under-exposure prevention measure . . . . .	204
<a href="#">OFIQ_LIB::modules::measures::UnifiedQualityScore</a>	
Implementation of the unified quality measure . . . . .	206



## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">mainpage.h</a>	This header file is for generating the doxygen documentation for OFIQ . . . . .	209
<a href="#">ofiq_lib.h</a>	Class describing the interface to the OFIQ . . . . .	210
<a href="#">ofiq_lib_impl.h</a>	Implementation of the OFIQ_LIB . . . . .	211
<a href="#">ofiq_structs.h</a>	PRovides several helper classes, enums and functions used in the OFIQ framework . . . . .	213
<a href="#">AllDetectors.h</a>	Provides the include statements to all classes derived from <a href="#">FaceDetectorInterface</a> . . . . .	218
<a href="#">detectors.h</a>	Provides the interface class to the face detector implementations . . . . .	219
<a href="#">opencv_ssd_face_detector.h</a>	Implementation of a face detector using the SSD face detector CNN . . . . .	220
<a href="#">adnet_FaceMap.h</a>	Provides definitions of landmark indices to access specific parts of ADNet landmarks . . . . .	221
<a href="#">adnet_landmarks.h</a>	Provides the <a href="#">ADNetFaceLandmarkExtractor</a> class . . . . .	224
<a href="#">AllLandmarks.h</a>	Provides the include statements to all classes derived from <a href="#">FaceLandmarkExtractorInterface</a> . . . . .	225
<a href="#">FaceMeasures.h</a>	Provides a class implementing two luminance measures . . . . .	226
<a href="#">FaceParts.h</a>	PRovides several helper classes, enums and functions used in the OFIQ framework . . . . .	228
<a href="#">landmarks.h</a>	Provides the base class for the implementation of face landmark extractors . . . . .	230
<a href="#">PartExtractor.h</a>	Provides helper class for face landmark handling . . . . .	231
<a href="#">AllMeasures.h</a>	Provides all classes derived from the <a href="#">OFIQ_LIB::modules::measures::Measure</a> class . . . . .	232
<a href="#">BackgroundUniformity.h</a>	Provides a class implementing the background uniformity measure . . . . .	234
<a href="#">CompressionArtifacts.h</a>	Provides a class implemtenting the no compression artifact measure . . . . .	235
<a href="#">CropOfTheFaceImage.h</a>	Provides a class implementing the crop of the face image measure . . . . .	237

<a href="#">DynamicRange.h</a>	Provides a class implementing the dynamic range measure . . . . .	238
<a href="#">Executor.h</a>	This class takes care of the computation of the measures activated . . . . .	239
<a href="#">ExpressionNeutrality.h</a>	Provides a class implementing the expression neutrality measure . . . . .	241
<a href="#">EyesOpen.h</a>	Provides a class implementing the eyes open measure . . . . .	242
<a href="#">EyesVisible.h</a>	Provides a class implementing the eyes visible measure . . . . .	244
<a href="#">FaceOcclusionPrevention.h</a>	Provides a class implementing the face occlusion prevention measure . . . . .	245
<a href="#">HeadPose.h</a>	Provides a class implementing head pose measures . . . . .	246
<a href="#">HeadSize.h</a>	Provides a class implementing the head size measure . . . . .	248
<a href="#">IlluminationUniformity.h</a>	Provides a class implementing the illumination uniformity measure . . . . .	249
<a href="#">InterEyeDistance.h</a>	Provides a class implementing the inter-eye distance measure . . . . .	250
<a href="#">Luminance.h</a>	Provides a class implementing two luminance measures . . . . .	252
<a href="#">Measure.h</a>	Provides the base class for all measures implemented in OFIQ . . . . .	253
<a href="#">MeasureFactory.h</a>	Provides a class for requesting creation of measure implementations . . . . .	255
<a href="#">MouthClosed.h</a>	Provides a class implementing the mouth closed measure . . . . .	257
<a href="#">MouthOcclusionPrevention.h</a>	Provides a class implementing the mouth occlusion prevention measure . . . . .	258
<a href="#">NaturalColour.h</a>	Provides a class implementing the natural colour measure . . . . .	259
<a href="#">NoHeadCoverings.h</a>	Provides a class implementing the no head covering measure . . . . .	261
<a href="#">OverExposurePrevention.h</a>	Provides a class implementing the background uniformity measure . . . . .	262
<a href="#">Sharpness.h</a>	Provides a class implementing the sharpness measure . . . . .	263
<a href="#">SingleFacePresent.h</a>	Provides a class implementing the single face present measure . . . . .	265
<a href="#">UnderExposurePrevention.h</a>	Provides a class implementing the under-exposure prevention measure . . . . .	266
<a href="#">UnifiedQualityScore.h</a>	Provides a class implementing the unified quality measure . . . . .	267
<a href="#">AllPoseEstimators.h</a>		269
<a href="#">HeadPose3DDFAV2.h</a>	Provides a class implementing a head pose estimator based on <a href="https://github.com/cleardusk/3DDFA_V2">https://github.com/cleardusk/3DDFA_V2</a> . . . . .	269
<a href="#">poseEstimators.h</a>	Base class for the different implementation of pose estimation algorithms . . . . .	271
<a href="#">FaceOcclusionSegmentation.h</a>	Provides a class for segmenting the facial part not occluded by any non-facial parts from an image . . . . .	273
<a href="#">FaceParsing.h</a>	Provides a class implementing the face parsing pre-processing . . . . .	274
<a href="#">ONNXRTSegmentation.h</a>	Helper class to manage the ONNXRuntime session object . . . . .	276

<a href="#">segmentations.h</a>	
Base class for the different implementation of segmentation algorithms . . . . .	277
<a href="#">Configuration.h</a>	
Provides a configuration class for handling configurations . . . . .	279
<a href="#">image_io.h</a>	
Provides helper functions for reading/writing images from/to disk . . . . .	281
<a href="#">image_utils.h</a>	
Provides image utility functions such as color conversion, luminance computation etc . . . . .	282
<a href="#">NeuronalNetworkContainer.h</a> . . . . .	284
<a href="#">OFIQError.h</a>	
Provides a class for the error handling within the QFIQ . . . . .	285
<a href="#">Session.h</a>	
The session class is the data container used to distribute the image and additional data, including the data computed during the pre-processing . . . . .	286
<a href="#">utils.h</a>	
Helper functions used by several classes . . . . .	288





## 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 describing bounding boxes, e.g. the face region of the faces found by a face detector.*
- struct [FacelImageQualityAssessment](#)  
*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 ,  
`CropOfTheFacelImage` = -0x54 , `LeftwardCropOfTheFacelImage` = 0x54 , `RightwardCropOfTheFacelImage` =  
0x55 , `MarginAboveOfTheFacelImage` = 0x56 ,  
`MarginBelowOfTheFacelImage` = 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
CropOfTheFacelImage	CropOfTheFacelImage: common measure for {Left,Right}wardCropOfTheFacelImage, MarginAbove, and MarginBelow
LeftwardCropOfTheFacelImage	LeftwardCropOfTheFacelImage
RightwardCropOfTheFacelImage	RightwardCropOfTheFacelImage
MarginAboveOfTheFacelImage	MarginAbove
MarginBelowOfTheFacelImage	MarginBelow
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 [NeuralNetworkContainer](#)  
*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 OFIQ::ReturnStatus readImageFromByteArray](#) (const std::vector< unsigned char > &buffer, [OFIQ::Image](#) &image)  
*Read image from byte array.*
- [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 &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 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).*
- [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](#) &exposureRange)  
*Helper function for some measures.*
- [OFIQ\\_EXPORT double ComputeBrightnessAspect](#) (const cv::Mat &luminanceImage, const cv::Mat &maskImage, 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` &`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 `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` `MakeGreylImage` (`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.

### 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

<i>faceImage</i>	Input image.
<i>faceLandmarks</i>	Face landmarks, based on the face represented in the input image.
<i>alignedFaceLandmarks</i>	Face landmarks of the aligned face image.
<i>transformationMatrix</i>	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

<i>session</i>	Session object containing the original facial image and pre-processing results computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method
<i>exposureRange</i>	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

<i>faceLandmarks</i>	Input face landmarks.
<i>leftEyeCenter</i>	Point coordinates of the left eye center.
<i>rightEyeCenter</i>	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	<i>landmarks</i>	Facial landmarks
out	<i>leftEyeCenter</i>	Left eye center computed from landmarks
out	<i>rightEyeCenter</i>	Right eye center computed from landmarks
out	<i>interEyeDistance</i>	Inter-eye distance computed from landmarks (does not consider the yaw angle).
out	<i>eyeMouthDistance</i>	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	<i>leftRegionOfInterest</i>	Rectangular region corresponding to the left eye
out	<i>rightRegionOfInterest</i>	Rectangular region corresponding to the right eye
in	<i>leftEyeCenter</i>	Center of the left eye
in	<i>rightEyeCenter</i>	Center of the right eye
in	<i>interEyeDistance</i>	Planar inter-eye distance
in	<i>eyeMouthDistance</i>	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

<i>v</i>	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

<i>luminanceImage</i>	luminance image.
<i>maskImage</i>	The mask on which the aspect is computed
<i>exposureRange</i>	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	<i>bgrImage</i>	BGR image
out	<i>a</i>	CIELAB value $a^*$
out	<i>b</i>	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

<i>sourceImage</i>	Input image.
<i>asGrayImage</i>	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

<i>x</i>	Argument
<i>k</i>	Argument
<i>eps</i>	Argument $\epsilon$

#### 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

<i>faceRects</i>	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**

<i>bgrImage</i>	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::Mat_1f & histogram )
```

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

**Parameters**

in	<i>luminanceImage</i>	Luminance image as returned by <a href="#">GetLuminanceImageFromBGR()</a> .
in	<i>maskImage</i>	The histogram is computed on pixels where the values of maskImage are non-zero.
out	<i>histogram</i>	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**

<i>width</i>	Width of the generated image.
<i>height</i>	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-squared bounding box into an squared one. The side length is defined by the greater one of height or width.

## Parameters

<i>i_bb</i>	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 squared image as input. This function consumes a bounding box and an input image. The greater parameter of width or height is used to define the side length of the new squared bounding box. The face will be centered in the bounding box. Padding is added if needed. The squared bounding box is used to generate a new cropped image, the `o_output_image`. Required translations are described by the translation vector `o_translation_vector`.

## Parameters

<i>i_bb</i>	Initial bounding box.
<i>i_input_image</i>	Input image.
<i>o_output_image</i>	Cropped output image. Cropping is based on the computed squared bounding box.
<i>o_bb</i>	Squared bounding box.
<i>o_translation_vector</i>	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	<i>filename</i>	Path and file name of the image being read from disk.
out	<i>image</i>	Reference to the image object where the data is loaded to.

## Returns

OFIQ::ReturnStatus

**6.3.3.18 readImageFromByteArray()**

```
OFIQ_EXPORT OFIQ::ReturnStatus OFIQ_LIB::readImageFromByteArray (
    const std::vector< unsigned char > & buffer,
    OFIQ::Image & image )
```

Read image from byte array.

## Parameters

in	<i>buffer</i>	Data as byte array of the image being read.
out	<i>image</i>	Reference to the image object where the data is loaded to.

## Returns

OFIQ::ReturnStatus

**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

<i>faceLandmarks</i>	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 openness.

### 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 openness.

### 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](#)  
*Implementation 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 [ExecutorLogActive](#) = 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

<i>msg</i>	Message to be logged.
------------	-----------------------

### 6.8.3 Variable Documentation

#### 6.8.3.1 ExecutorLogActive

```
const bool OFIQ_LIB::modules::measures::ExecutorLogActive = 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

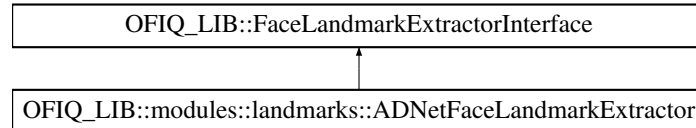
## Class Documentation

### 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:



#### 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

<i>config</i>	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

<i>session</i>	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::ADNetFaceLandmarkExtractor::landmarkExtractor_ [private]
```

Encapsulated implementation class.

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

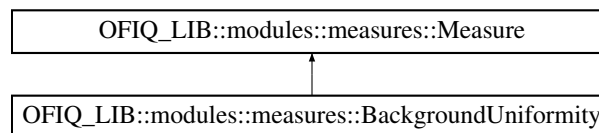
- [adnet\\_landmarks.h](#)

**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:

**Public Member Functions**

- [BackgroundUniformity](#) (const [Configuration](#) &configuration)  
*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, [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\\_targetHeight](#) = 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\\_targetWidth](#) = 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\\_cropLeft](#) = 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\\_cropRight](#) = 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\\_cropTop](#) = 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\\_cropBottom](#) = 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\\_erosionKernelSize](#) = 4  
*Size of the erosion kernel applied to the background as per [OFIQ\\_LIB::modules::segmentations::FaceParsing](#) to reduce the risk that background uniformity 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 background 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 ) [explicit]
```

Constructor.

Parameters

<i>configuration</i>	Configuration object from which the measure related configuration is read.
----------------------	--

## 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

<i>session</i>	Session object computed by the <a href="#">OFIQImpl::performPreprocessing()</a> .
----------------	---

Implements [OFIQ\\_LIB::modules::measures::Measure](#).

## 7.2.4 Member Data Documentation

### 7.2.4.1 m\_cropBottom

```
uint16_t OFIQ_LIB::modules::measures::BackgroundUniformity::m_cropBottom = 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\_cropLeft

```
uint16_t OFIQ_LIB::modules::measures::BackgroundUniformity::m_cropLeft = 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\_cropRight

```
uint16_t OFIQ_LIB::modules::measures::BackgroundUniformity::m_cropRight = 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\_cropTop

```
uint16_t OFIQ_LIB::modules::measures::BackgroundUniformity::m_cropTop = 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\_erosionKernelSize

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

Size of the erosion kernel applied to the background as per [OFIQ\\_LIB::modules::segmentations::FaceParsing](#) to reduce the risk that background uniformity estimation is applied to part of the subject.

#### 7.2.4.6 m\_targetHeight

```
uint16_t OFIQ_LIB::modules::measures::BackgroundUniformity::m_targetHeight = 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\_targetWidth

```
uint16_t OFIQ_LIB::modules::measures::BackgroundUniformity::m_targetWidth = 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

<i>xleft</i>	x coordinate of the upper left point of the bounding box.
<i>ytop</i>	y coordinate of the upper left point of the bounding box.
<i>width</i>	width of the bounding box.
<i>height</i>	height of the bounding box.
<i>i_faceDetector</i>	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:

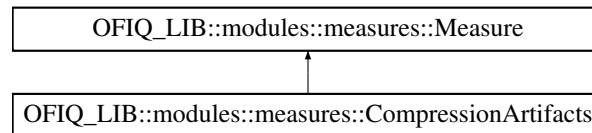
- [ofiq\\_structs.h](#)

## 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:



### Public Member Functions

- [CompressionArtifacts](#) (const [Configuration](#) &configuration)  
*Constructor.*
- void [Execute](#) ([OFIQ\\_LIB::Session](#) &session) override  
*Assesses absence of compression artifacts.*

### Public Member Functions inherited from [OFIQ\\_LIB::modules::measures::Measure](#)

- [Measure](#) (const [Configuration](#) &configuration, [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

- int [m\\_crop](#)  
*Top, right, left, and bottom margin by which the aligned image is cropped.*
- int [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 absence 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 ) [explicit]
```

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

<i>configuration</i>	Configuration object from which measure-related configuration is read.
----------------------	--

## Exceptions

<i>OFIQ_LIB::OFIQError</i>	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 absence of compression artifacts.

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

## Parameters

<i>session</i>	Session object computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method.
----------------	---

Implements [OFIQ\\_LIB::modules::measures::Measure](#).

### 7.4.4 Member Data Documentation

#### 7.4.4.1 m\_crop

```
int 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

```
int 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\_view 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

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

## 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

<i>key</i>	Key of the configuration.
------------	---------------------------

#### Returns

The accessed boolean configuration.

#### Exceptions

<i>OFIQ_LIB::OFIQError</i>	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

<i>key</i>	Key of the configuration.
<i>value</i>	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

<i>key</i>	Key of the configuration.
------------	---------------------------

#### Returns

The accessed double configuration.

#### Exceptions

<i>OFIQ_LIB::OFIQError</i>	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

<i>key</i>	Key of the configuration.
<i>value</i>	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

<i>key</i>	Key of the configuration.
------------	---------------------------

#### Returns

The accessed string configuration.

#### Exceptions

<i>OFIQ_LIB::OFIQError</i>	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

<i>key</i>	Key of the configuration.
<i>value</i>	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**

<i>key</i>	Key of the configuration.
<i>value</i>	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_view dataDir )
```

Sets the configuration directory.

**Parameters**

<i>dataDir</i>	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:

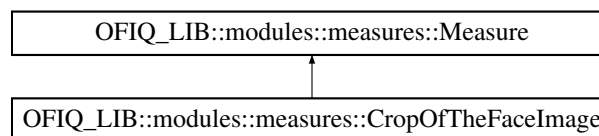
- [Configuration.h](#)

## 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:



### Public Member Functions

- [CropOfTheFaceImage](#) (const [Configuration](#) &configuration)  
*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, [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 ) [explicit]
```

Constructor.

Parameters

<i>configuration</i>	Configuration object from which measure-related configuration is read.
----------------------	--

### 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

<i>session</i>	Session object.
----------------	-----------------

Implements [OFIQ\\_LIB::modules::measures::Measure](#).

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

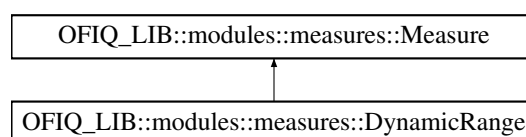
- [CropOfTheFacelImage.h](#)

## 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:



#### Public Member Functions

- [DynamicRange](#) (const [Configuration](#) &configuration)  
*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, [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 ) [explicit]
```

Constructor.

## Parameters

<i>configuration</i>	Configuration object from which the measure-related configuration is read.
----------------------	--

## 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

<i>session</i>	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](#) > > [m\\_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

<i>measures</i>	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>i_currentSession</i>	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 m\_measures

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

Container for access to the measures to be computed.

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

- [Executor.h](#)

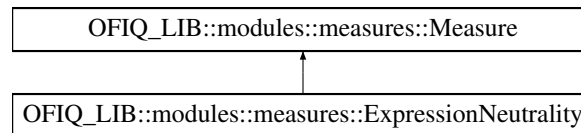


## 7.9 OFIQ\_LIB::modules::measures::ExpressionNeutrality Class Reference

Provides a class implementing the expression neutrality measure.

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

Inheritance diagram for OFIQ\_LIB::modules::measures::ExpressionNeutrality:



### Public Member Functions

- [ExpressionNeutrality](#) (const [Configuration](#) &configuration)  
*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, [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 > [m\\_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 ) [explicit]
```

Construct a new Expression Neutrality object.

#### Parameters

<i>configuration</i>	Configuration object from which measure-related configuration is read.
----------------------	--

## 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

<i>session</i>	Session object
----------------	----------------

Implements [OFIQ\\_LIB::modules::measures::Measure](#).

## 7.9.4 Member Data Documentation

### 7.9.4.1 m\_classifier

```
std::shared_ptr<cv::ml::Boost> OFIQ_LIB::modules::measures::ExpressionNeutrality::m_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:

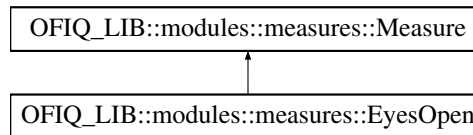
- [ExpressionNeutrality.h](#)

## 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:



### Public Member Functions

- [EyesOpen](#) (const [Configuration](#) &configuration)  
*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, [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 ) [explicit]
```

Constructor.

Parameters

<a href="#">configuration</a>	Configuration object from which measure-related configuration is read.
-------------------------------	--

### 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

<a href="#">session</a>	Session object computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method.
-------------------------	---

See also

[Session::getAlignedFaceLandmarks\(\)](#)

Implements [OFIQ\\_LIB::modules::measures::Measure](#).

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

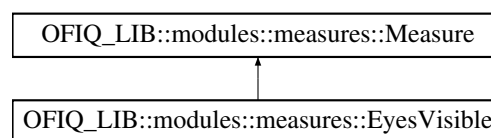
- [EyesOpen.h](#)

## 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:



### Public Member Functions

- [EyesVisible](#) (const [Configuration](#) &configuration)  
*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, [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 ) [explicit]
```

Constructor.

Parameters

<i>configuration</i>	Configuration object from which measure-related configuration is read.
----------------------	--

### 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

<code>session</code>	Session object computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method.
----------------------	---

Implements [OFIQ\\_LIB::modules::measures::Measure](#).

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

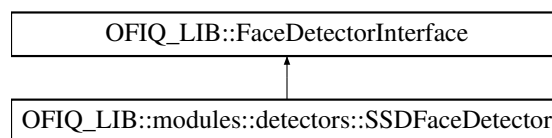
- [EyesVisible.h](#)

## 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	<i>session</i>	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

<i>session</i>	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 constructor.*
- [FaceImageQualityAssessment](#) (const [QualityAssessments](#) &qAssessments, const [BoundingBox](#) &boundingBox)  
*Parameterized constructor.*

### Public Attributes

- [QualityAssessments](#) qAssessments  
*Container for storing the results 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 constructor.

#### 7.13.2.2 FaceImageQualityAssessment() [2/2]

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

Parameterized constructor.

#### Parameters

in	<i>qAssessments</i>	
in	<i>boundingBox</i>	

### 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 results of the different measure computations.

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

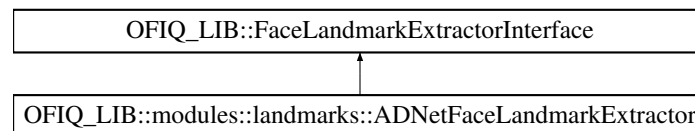
- [ofiq\\_structs.h](#)

## 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

<i>session</i>	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

<i>session</i>	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

<i>pair</i>	The two landmark points stored in the member <code>pair.Lower</code> and <code>pair.Upper</code> .
-------------	--

## 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

<i>a</i>	First landmark point
<i>b</i>	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

<i>faceLandmarks</i>	Facial landmarks object
<i>height</i>	Height of the mask image
<i>width</i>	Width of the mask image
<i>alpha</i>	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

<i>landmarks</i>	Facial landmarks
<i>facePart</i>	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

<i>pair</i>	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

<i>landmarks</i>	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**

<i>pairs</i>	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**

<i>faceLandmarks</i>	Facial landmarks
<i>yaw</i>	Yaw angle in degree

**Returns**

The inter-eye distance

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

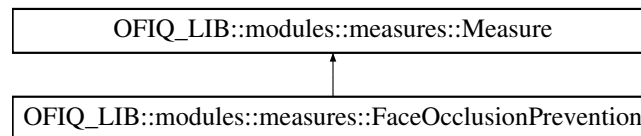
- [FaceMeasures.h](#)

## 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:



### Public Member Functions

- [FaceOcclusionPrevention](#) (const [Configuration](#) &configuration)  
*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, 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 ) [explicit]
```

Constructor.

Parameters

<i>configuration</i>	Configuration object from which measure-related configuration is read.
----------------------	--

### 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

<i>session</i>	Session object computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method.
----------------	---

See also

[FaceOcclusionSegmentation](#)

Implements [OFIQ\\_LIB::modules::measures::Measure](#).

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

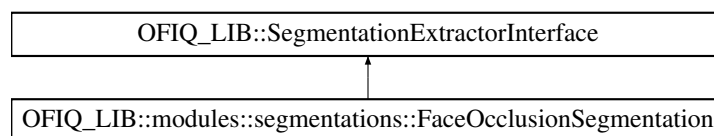
- [FaceOcclusionPrevention.h](#)

## 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:



### 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 > m_segmentationImage`  
*Stores the last result computed with [UpdateMask\(\)](#).*
- `const std::string m_modelConfigItem = "params.measures.FaceOcclusionSegmentation.model_path"`  
*JSON/JAXN key to access path to FaceExtraction's model file from [Configuration](#) object.*
- `const int m_cropLeft = 96`  
*Cropping parameter.*
- `const int m_cropRight = 96`  
*Cropping parameter.*
- `const int m_cropTop = 96`  
*Cropping parameter.*
- `const int m_cropBottom = 96`  
*Cropping parameter.*
- `const int m_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 m_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**

<i>config</i>	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**

<i>alignedImage</i>	Aligned image of dimension 616 x 616 as returned by <a href="#">Session::getAlignedFace()</a> .
---------------------	---

**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

<i>session</i>	Session object containing the original facial image and pre-processing results computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method.
<i>faceSegment</i>	Should be the value <a href="#">SegmentClassLabels::face</a> .

## Returns

Face occlusion segmentation mask.

Implements [OFIQ\\_LIB::SegmentationExtractorInterface](#).

## 7.18.4 Member Data Documentation

### 7.18.4.1 m\_cropBottom

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

Cropping parameter.

### 7.18.4.2 m\_cropLeft

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

Cropping parameter.

### 7.18.4.3 m\_cropRight

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

Cropping parameter.

### 7.18.4.4 m\_cropTop

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

Cropping parameter.

### 7.18.4.5 m\_modelConfigItem

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

JSON/JAXN key to access path to FaceExtraction's model file from [Configuration](#) object.

#### 7.18.4.6 m\_onnxRuntimeEnv

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

Manages CNN computations.

#### 7.18.4.7 m\_scaledHeight

```
const int OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation::m_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 m\_scaledWidth

```
const int OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation::m_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 m\_segmentationImage

```
std::shared_ptr<cv::Mat> OFIQ_LIB::modules::segmentations::FaceOcclusionSegmentation::m_↔
segmentationImage [private]
```

Stores the last result computed with [UpdateMask\(\)](#).

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

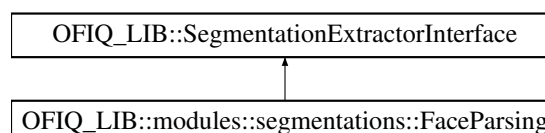
- [FaceOcclusionSegmentation.h](#)

## 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:





### 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](#) (const [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 > m\_segmentationImage  
*Stores the last result computed with [UpdateMask](#)() .*
- const std::string m\_modelConfigItem = "params.measures.FaceParsing.model\_path"  
*JSON/JAXN key to access path to [BiSeNet](#) model in ONNX format from [Configuration](#) object.*
- const int m\_imageSize = 400  
*Face parsing target dimension.*
- const int m\_cropLeft = 30  
*Cropping parameter.*
- const int m\_cropRight = 30  
*Cropping parameter.*
- const int m\_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

<i>config</i>	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

<i>resultImage</i>	Blob being created by one of the CreateBlob functions.
<i>i_imageSize_one_dim</i>	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

<i>image</i>	Input image
<i>i_imageSize_one_dim</i>	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 (
    const 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

<i>session</i>	Session object containing the original facial image and pre-processing results computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method.
<i>faceSegment</i>	Enum value encoding the requested face segment. If the value is <a href="#">SegmentClassLabels::face</a> then an image of dimension 400 x 400 is returned assigning each pixel a value between 0 and 18 as in the table of <a href="#">FaceParsing</a> ; 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 m\_cropBottom

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

Cropping parameter.

### 7.19.4.2 m\_cropLeft

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

Cropping parameter.

#### 7.19.4.3 m\_cropRight

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

Cropping parameter.

#### 7.19.4.4 m\_imageSize

```
const int OFIQ_LIB::modules::segmentations::FaceParsing::m_imageSize = 400 [private]
```

Face parsing target dimension.

#### 7.19.4.5 m\_modelConfigItem

```
const std::string OFIQ_LIB::modules::segmentations::FaceParsing::m_modelConfigItem = "params.↔
measures.FaceParsing.model_path" [private]
```

JSON/JAXN key to access path to [BiSeNet](#) model in ONNX format from [Configuration](#) object.

#### 7.19.4.6 m\_onnxRuntimeEnv

```
ONNXRuntimeSegmentation OFIQ_LIB::modules::segmentations::FaceParsing::m_onnxRuntimeEnv [private]
```

Manages CNN computations.

#### 7.19.4.7 m\_segmentationImage

```
std::shared_ptr<cv::Mat> OFIQ_LIB::modules::segmentations::FaceParsing::m_segmentationImage
[private]
```

Stores the last result computed with [UpdateMask\(\)](#).

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

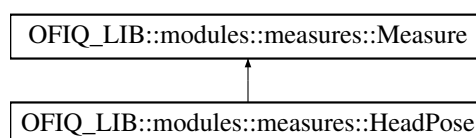
- [FaceParsing.h](#)

## 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:



## Public Member Functions

- [HeadPose](#) (const [Configuration](#) &configuration)  
*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, [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 ) [explicit]
```

Constructor for HeadPose.

Parameters

<i>configuration</i>	Configuration object from which measure-related configuration is read.
----------------------	--

### 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

<i>session</i>	Session object computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method.
----------------	---

Implements [OFIQ\\_LIB::modules::measures::Measure](#).

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

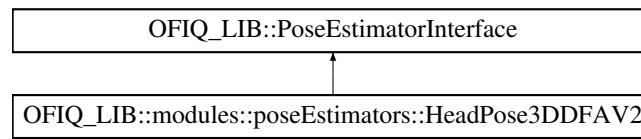
- [HeadPose.h](#)

## 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:



### 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) const  
*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\\_ortSession](#)  
*ONNXRuntime session handle.*
- int64\_t [m\\_expectedImageWidth](#) = 0  
*Width of the CNN used for computation, read from the loaded model.*
- int64\_t [m\\_expectedImageHeight](#) = 0  
*Height of the CNN used for computation, read from the loaded model.*
- int64\_t [m\\_expectedImageNumberOfChannels](#) = 0  
*Expected number of channels of the input image, read from the loaded model.*
- int64\_t [m\\_numberOfInputElements](#) = 0  
*Number of input elements of the CNN used for computation, read from the loaded model.*
- std::array< int64\_t, 4 > [m\\_inputShape](#)  
*inputShape of the CNN used for computation, read from the loaded model.*



## Static Private Attributes

- static const std::string [m\\_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 based on a CNN from [https://github.com/cleardusk/3DDFA\\_V2](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

<i>config</i>	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 ) const [private]
```

Crop face from image. Internally the passed bounding box will be transformed to a square region.

**Parameters**

<i>image</i>	Input image.
<i>biggestFace</i>	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**

<i>session</i>	Session object containing the original facial image and pre-processing results computed.
<i>pose</i>	Estimated head pose.

Implements [OFIQ\\_LIB::PoseEstimatorInterface](#).

**7.21.4 Member Data Documentation****7.21.4.1 m\_expectedImageHeight**

```
int64_t OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::m_expectedImageHeight = 0 [private]
```

Height of the CNN used for computation, read from the loaded model.

**7.21.4.2 m\_expectedImageNumberOfChannels**

```
int64_t OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::m_expectedImageNumberOfChannels =
0 [private]
```

Expected number of channels of the input image, read from the loaded model.

**7.21.4.3 m\_expectedImageWidth**

```
int64_t OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::m_expectedImageWidth = 0 [private]
```

Width of the CNN used for computation, read from the loaded model.

#### 7.21.4.4 m\_inputShape

```
std::array<int64_t, 4> OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::m_inputShape [private]
```

inputShape of the CNN used for computation, read from the loaded model.

#### 7.21.4.5 m\_numberOfInputElements

```
int64_t OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::m_numberOfInputElements = 0 [private]
```

Number of input elements of the CNN used for computation, read from the loaded model.

#### 7.21.4.6 m\_ortenv

```
Ort::Env OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::m_ortenv [private]
```

ONNXRuntime environment handle.

#### 7.21.4.7 m\_ortSession

```
std::unique_ptr<Ort::Session> OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::m_ortSession [private]
```

ONNXRuntime session handle.

#### 7.21.4.8 m\_paramPoseEstimatorModel

```
const std::string OFIQ_LIB::modules::poseEstimators::HeadPose3DDFAV2::m_paramPoseEstimatorModel [static], [private]
```

Name of the used CNN net, passed from the configuration.

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

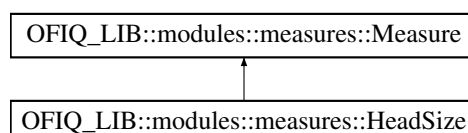
- [HeadPose3DDFAV2.h](#)

## 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:



## Public Member Functions

- [HeadSize](#) (const [Configuration](#) &configuration)  
*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, [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 ) [explicit]
```

Constructor.

Parameters

<i>configuration</i>	Configuration object from which measure-related configuration is read.
----------------------	--

### 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

<i>session</i>	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:

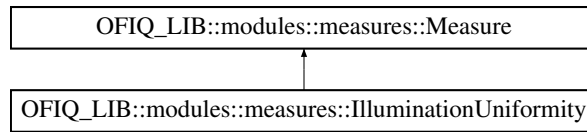
- [HeadSize.h](#)

## 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)  
*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, [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 ) [explicit]
```

Constructor.

Parameters

<a href="#">configuration</a>	Configuration object from which measure-related configuration is read.
-------------------------------	--

### 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

<a href="#">session</a>	Session object computed by the <a href="#">OFIQImpl::performPreprocessing()</a> 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](#), const std::shared\_ptr< uint8\_t > &[data](#))  
*Constructor.*
- size\_t [size](#) () const  
*This function returns the size of the image data.*
- void [deepcopy](#) (uint16\_t [width](#), uint16\_t [height](#), uint8\_t [depth](#), const std::shared\_ptr< uint8\_t > &[data](#))  
*Overwrites the data of the image being a deepcopy of the specified parameters.*

### 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,
    const std::shared_ptr< uint8_t > & data ) [inline]
```

Constructor.

#### Attention

The constructor sets a shallow copy of the specified data pointer. Therefore, this constructor should no be used from binding such as Java/JNI using byte data of which memory is managed by another mechanism. Rather, for a binding such as Java/JNI, one should use the [deepcopy](#) method.



## Parameters

<i>width</i>	of the image.
<i>height</i>	of the image.
<i>depth</i>	of the image
<i>data</i>	of the image.

## 7.24.3 Member Function Documentation

### 7.24.3.1 deepcopy()

```
void OFIQ::Image::deepcopy (
    uint16_t width,
    uint16_t height,
    uint8_t depth,
    const std::shared_ptr< uint8_t > & data ) [inline]
```

Overwrites the data of the image being a deepcopy of the specified parameters.

This method can be used by a binding such as Java/JNI when the memory of the specified data is managed by another mechanism such as Java's garbage collector.

## Parameters

in	<i>width</i>	of the image.
in	<i>height</i>	of the image.
in	<i>depth</i>	of the image
in	<i>data</i>	of the image.

### 7.24.3.2 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:

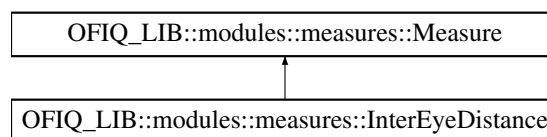
- [ofiq\\_structs.h](#)

## 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:



### Public Member Functions

- [InterEyeDistance](#) (const [Configuration](#) &configuration)  
*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, [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 ) [explicit]
```

Constructor.

## Parameters

<i>configuration</i>	Configuration object from which measure-related configuration is read.
----------------------	--

## 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

<i>session</i>	Session object computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method.
----------------	---

Implements [OFIQ\\_LIB::modules::measures::Measure](#).

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

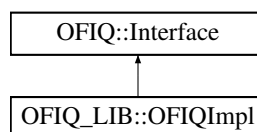
- [InterEyeDistance.h](#)

## 7.26 OFIQ::Interface Class Reference

The interface to FACE QA implementation.

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

Inheritance diagram for OFIQ::Interface:



### 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.*
- [OFIQ\\_EXPORT](#) void [getVersion](#) (int &major, int &minor, int &patch) const

## 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 getVersion()

```
OFIQ_EXPORT void OFIQ::Interface::getVersion (
    int & major,
    int & minor,
    int & patch ) const
```

#### 7.26.3.3 initialize()

```
virtual OFIQ::ReturnStatus OFIQ::Interface::initialize (
    const std::string & configDir,
    const std::string & configFileNames ) [pure virtual]
```

This function initializes the implementation under test. The implementation under test should set all parameters.

**Parameters**

in	<i>configDir</i>	string representation of the directory containing the configuration file specified by configFileName
in	<i>configFileName</i>	An string value encoding the JAXN configuration file name

**Returns**

OFIQ::ReturnStatus indicating if the initialization was successful.

Implemented in [OFIQ\\_LIB::OFIQImpl](#).

**7.26.3.4 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	<i>face</i>	Single face image
out	<i>quality</i>	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](#).

**7.26.3.5 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	<i>image</i>	Single face image
out	<i>assessments</i>	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	<i>upper</i>	LandmarkPoint of first landmark.
in	<i>lower</i>	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

$i_x$ _ $x$	x - coordinate of the landmark.
$i_y$ _ $y$	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:

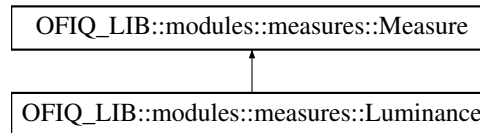
- [ofiq\\_structs.h](#)

## 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:



### Public Member Functions

- [Luminance](#) (const [Configuration](#) &configuration)  
*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, [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 ) [explicit]
```

Constructor.

Parameters

<i>configuration</i>	Configuration object from which measure-related configuration is read.
----------------------	--

### 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

<i>session</i>	Session object computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method.
----------------	---

Implements [OFIQ\\_LIB::modules::measures::Measure](#).

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

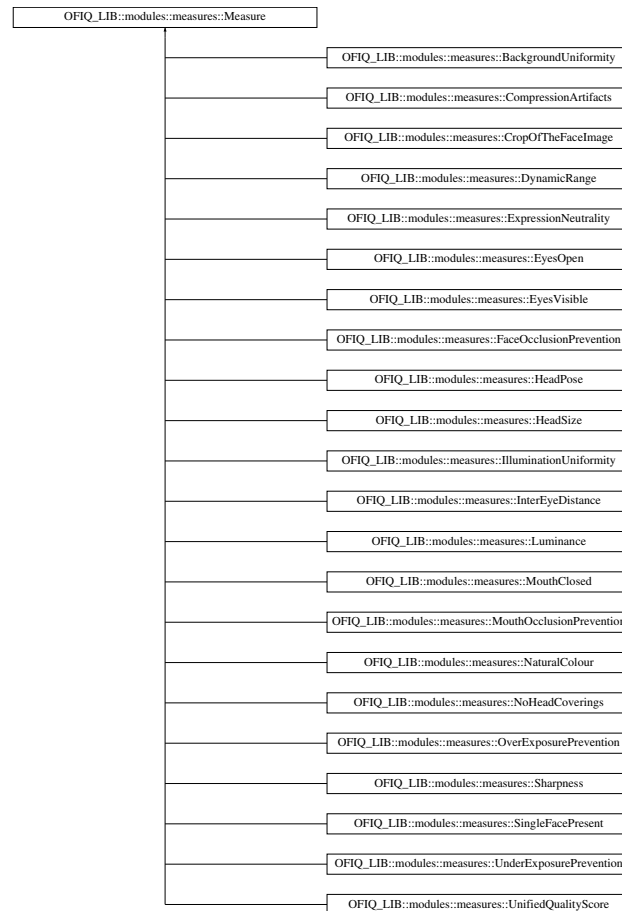
- [Luminance.h](#)

## 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, [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<> > [m\\_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,
    OFIQ::QualityMeasure measure ) [inline]
```

Constructor.

## Parameters

<i>configuration</i>	Configuration object from which measure-related configuration is read.
<i>session</i>	Session object containing the original facial image and pre-processing results computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method
<i>measure</i>	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

<i>key</i>	Key/name of the measure of which mapping is configured.
<i>defaultValues</i>	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

<i>measure</i>	Enum value encoding the measure for which the mapping is configured.
<i>defaultValues</i>	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

<i>session</i>	Session object containing the original facial image and pre-processing results computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method.
----------------	---

Implemented in [OFIQ\\_LIB::modules::measures::BackgroundUniformity](#), [OFIQ\\_LIB::modules::measures::CompressionArtifacts](#), [OFIQ\\_LIB::modules::measures::CropOfTheFacelImage](#), [OFIQ\\_LIB::modules::measures::DynamicRange](#), [OFIQ\\_LIB::modules::measures::EyesOpen](#), [OFIQ\\_LIB::modules::measures::EyesVisible](#), [OFIQ\\_LIB::modules::measures::FaceOcclusion](#), [OFIQ\\_LIB::modules::measures::HeadPose](#), [OFIQ\\_LIB::modules::measures::HeadSize](#), [OFIQ\\_LIB::modules::measures::Illumination](#), [OFIQ\\_LIB::modules::measures::InterEyeDistance](#), [OFIQ\\_LIB::modules::measures::Luminance](#), [OFIQ\\_LIB::modules::measures::MouthOcclusion](#), [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::UnifiedQualityScore](#) 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

<i>key</i>	Key/name of the measure used to read parameters from a private map member.
<i>rawValue</i>	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**

<i>measure</i>	Enum value of the measure used to read parameters from a private map member.
<i>rawValue</i>	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**

<i>rawKey</i>	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**

<i>measure</i>	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**

<i>rawValue</i>	Native quality score.
<i>par</i>	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

<i>session</i>	Session object containing the original facial image and pre-processing results computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method.
<i>measure</i>	Enum value specifying this measure.
<i>rawValue</i>	Native quality score
<i>code</i>	Value indicating whether the quality assessment was computed successfully or otherwise (e.g., <code>failureToAssess</code> ).

**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

<i>x</i>	Native quality score
<i>x0</i>	Non-zero center point
<i>w</i>	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 m\_sigmoidMap

```
std::map<std::string, SigmoidParameters, std::less<> > OFIQ_LIB::modules::measures::MeasureFactory::m_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)

*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 ) [static]
```

Requests the creation of a measure implementation.

## Parameters

<i>measure</i>	Enum value encoding the requested measure.
<i>configuration</i>	Configuration from which measure parameters are read.

## 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:

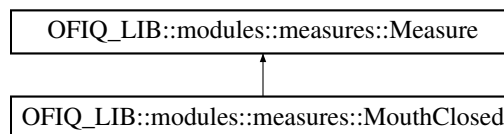
- [MeasureFactory.h](#)

## 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:



### Public Member Functions

- [MouthClosed](#) (const [Configuration](#) &configuration)  
*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, [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 ) [explicit]
```

Constructor.

Parameters

<i>configuration</i>	Configuration object from which measure-related configuration is read.
----------------------	--

### 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

<i>session</i>	Session object computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method.
----------------	---

See also

[Session::getAlignedFaceLandmarks\(\)](#)

Implements [OFIQ\\_LIB::modules::measures::Measure](#).

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

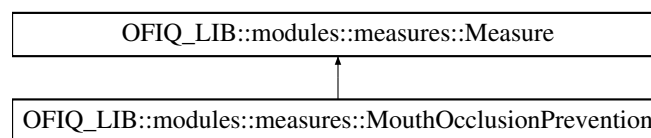
- [MouthClosed.h](#)

## 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)  
*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, [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 ) [explicit]
```

Constructor.

## Parameters

<i>configuration</i>	Configuration object from which measure-related configuration is read.
----------------------	--

### 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

<i>session</i>	Session object computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method.
----------------	---

## See also

[FaceOcclusionSegmentation](#)

[Session::getAlignedFaceLandmarks\(\)](#)

Implements [OFIQ\\_LIB::modules::measures::Measure](#).

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

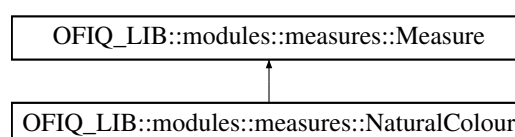
- [MouthOcclusionPrevention.h](#)

## 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`:





## Public Member Functions

- [NaturalColour](#) (const [Configuration](#) &configuration)  
*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, [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) const  
*Creates a mask image from the convex full of the specified landmarks.*
- cv::Mat [ReduceImageToRegionOfInterest](#) (const cv::Mat &maskedImage, const cv::Rect &leftRegionOfInterest, const cv::Rect &rightRegionOfInterest) const  
*Extracts two rectangular regions from an image and returns its concatenation.*
- double [CalculateScore](#) (double meanChannelA, double meanChannelB) const  
*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 ) [explicit]
```

Constructor.

Parameters

<i>configuration</i>	Configuration object from which measure-related configuration is read.
----------------------	--

### 7.34.3 Member Function Documentation

#### 7.34.3.1 CalculateScore()

```
double OFIQ_LIB::modules::measures::NaturalColour::CalculateScore (
    double meanChannelA,
    double meanChannelB ) const [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

<i>meanChannelA</i>	The CIELAB value $a^*$ input value.
<i>meanChannelB</i>	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 ) const [private]
```

Creates a mask image from the convex full of the specified landmarks.

## Parameters

<i>landmarks</i>	Facial landmarks.
<i>cvImage</i>	The mask image returned has the same dimension as <i>cvImage</i> .

## 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

<i>session</i>	Session object computed by the <a href="#">OFIQImpl::performPreprocessing()</a> 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 ) const [private]
```

Extracts two rectangular regions from an image and returns its concatenation.

## Parameters

<i>maskedImage</i>	The input image from which the two regions are extracted.
<i>leftRegionOfInterest</i>	First region
<i>rightRegionOfInterest</i>	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< PoseEstimatorInterface > poseEstimator`  
*Pointer to a [SegmentationExtractorInterface](#) .*
- `std::shared_ptr< SegmentationExtractorInterface > faceOcclusionExtractor`  
*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

<i>faceDetector</i>	Implementation of a <a href="#">FaceDetectorInterface</a>
<i>landmarkExtractor</i>	Implementation of a <a href="#">FaceLandmarkExtractorInterface</a>
<i>segmentationExtractor</i>	Implementation of a <a href="#">SegmentationExtractorInterface</a> . A pointer to an object instantiated from the <a href="#">FaceParsing</a> class would be valid.
<i>poseEstimator</i>	Implementation of a <a href="#">PoseEstimatorInterface</a>
<i>faceOcclusionExtractor</i>	Implementation of a <a href="#">SegmentationExtractorInterface</a> . A pointer to an object instantiated from the <a href="#">FaceOcclusionSegmentation</a> 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::faceOcclusionExtractor
```

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::landmarkExtractor
```

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::segmentationExtractor
```

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:

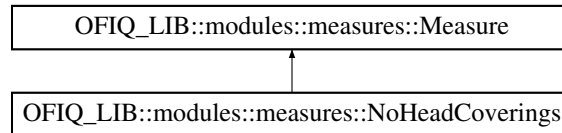
- [NeuronalNetworkContainer.h](#)

## 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:



### Public Member Functions

- [NoHeadCoverings](#) (const [Configuration](#) &configuration)  
*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, [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 [m\\_t0](#)  
*Lower threshold.*
- double [m\\_t1](#)  
*Upper threshold.*
- double [m\\_w](#)  
*Standard deviation used in sigmoid function.*
- double [m\\_x0](#)  
*Development point used in sigmoid function.*

## 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 ) [explicit]
```

Constructor.

The configuration object can optionally configure the threshold using the `params.measures.NoHeadCoverings.threshold` key which is 0.02 by default.

## Parameters

<i>configuration</i>	Configuration object from which measure-related configuration is read.
----------------------	--

### 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

<i>session</i>	Session object computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method.
----------------	---

## See also

[FaceParsing](#)

Implements [OFIQ\\_LIB::modules::measures::Measure](#).

### 7.36.4 Member Data Documentation

#### 7.36.4.1 m\_t0

```
double OFIQ_LIB::modules::measures::NoHeadCoverings::m_t0 [private]
```

Lower 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;

#### 7.36.4.2 m\_t1

```
double OFIQ_LIB::modules::measures::NoHeadCoverings::m_t1 [private]
```

Upper 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 0 (worst) is used;



### 7.36.4.3 m\_w

```
double OFIQ_LIB::modules::measures::NoHeadCoverings::m_w [private]
```

Standard deviation used in sigmoid function.

If the native quality score is between (m\_t0,m\_t1), then the quality component value is interpolated using a sigmoid function with a standard deviation specified by m\_w.

### 7.36.4.4 m\_x0

```
double OFIQ_LIB::modules::measures::NoHeadCoverings::m_x0 [private]
```

Development point used in sigmoid function.

If the native quality score is between (m\_t0,m\_t1), then the quality component value is interpolated using a sigmoid function with a development point specified by m\_x0.

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

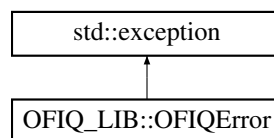
- [NoHeadCoverings.h](#)

## 7.37 OFIQ\_LIB::OFIQError Class Reference

Implementation of a custom exception.

```
#include <OFIQError.h>
```

Inheritance diagram for OFIQ\_LIB::OFIQError:



### Public Member Functions

- [OFIQError](#) ([OFIQ::ReturnCode](#) returnCode, std::string\_view message)  
*Constructor.*
- 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 m\\_returnCode](#)  
*Member storing the ReturnCode.*
- `std::string m_message`  
*Member, storing the message passed in the constructor.*
- `std::string m_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 )
```

Constructor.

#### Parameters

<i>returnCode</i>	Based on the <a href="#">OFIQ::ReturnCode</a> (see <a href="#">OFIQ::ReturnCode</a> ).
<i>message</i>	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 m\_extendedMessage

```
std::string OFIQ_LIB::OFIQError::m_extendedMessage [private]
```

The extended message merges the ReturnCode and the message into one string.

### 7.37.4.2 m\_message

```
std::string OFIQ_LIB::OFIQError::m_message [private]
```

Member, storing the message passed in the constructor.

### 7.37.4.3 m\_returnCode

```
OFIQ::ReturnCode OFIQ_LIB::OFIQError::m_returnCode [private]
```

Member storing the ReturnCode.

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

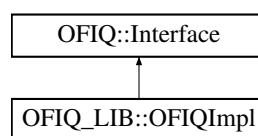
- [OFIQError.h](#)

## 7.38 OFIQ\_LIB::OFIQImpl Class Reference

Implementation of the OFIQ\_LIB.

```
#include <ofiq_lib_impl.h>
```

Inheritance diagram for OFIQ\_LIB::OFIQImpl:



### 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::FacelImageQualityAssessment](#) &assessments) override  
*Run the computation of all measures set in the configuration.*

## Public Member Functions inherited from OFIQ::Interface

- virtual [~Interface](#) ()=default  
*Default Destructor.*
- [OFIQ\\_EXPORT](#) void [getVersion](#) (int &major, int &minor, int &patch) const

## Private Member Functions

- std::unique\_ptr< [OFIQ\\_LIB::modules::measures::Executor](#) > [CreateExecutor](#) ()  
*Create a Executor object.*
- void [CreateNetworks](#) ()  
*Create a NeuronalNetworkContainer.*
- void [performPreprocessing](#) ([Session](#) &session)  
*Perform the preprocessing.*
- void [alignFacelImage](#) ([Session](#) &session) const  
*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::FacelImageQualityAssessment](#) [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 ) const [private]
```

Perform the face alignment.

#### Parameters

<i>session</i>	Session object containing the original facial image and pre-processing results computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method
----------------	--

### 7.38.3.2 CreateExecutor()

```
std::unique_ptr< OFIQ_LIB::modules::measures::Executor > OFIQ_LIB::OFIQImpl::CreateExecutor (
) [private]
```

Create a Executor object.

#### 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

<i>configDir</i>	Path to the configuration file.
<i>configValue</i>	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

<i>session</i>	Session object containing the original facial image and pre-processing results computed by the <a href="#">OFIQImpl::performPreprocessing()</a> 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	<i>face</i>	Input image.
out	<i>quality</i>	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	<i>image</i>	Input image.
out	<i>assessments</i>	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 preprocessing.

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](#) () const  
*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\\_memoryInfo](#) = 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\\_ortSession](#)  
*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 ( ) const
```

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>i_model_data</i>	Model data loaded from file.
<i>i_imageWidth</i>	Width of the input image as expected by the model.
<i>i_imageHeight</i>	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>i_modelData</i>	Model data loaded from file.
<i>i_imageWidth</i>	Width of the input image as expected by the model.
<i>i_imageHeight</i>	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>i_netInput</i>	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\_memoryInfo

```
Ort::MemoryInfo ONNXRuntimeSegmentation::m_memoryInfo = Ort::MemoryInfo::CreateCpu(OrtDevice←
Allocator, OrtMemTypeCPU) [private]
```

ONNXRuntime variable to setup the tensors used in ONNXRuntime.

### 7.39.4.3 m\_ortenv

```
Ort::Env ONNXRuntimeSegmentation::m_ortenv [private]
```

Handle to the ONNXRuntime environment variable.

#### 7.39.4.4 m\_ortSession

```
std::unique_ptr<Ort::Session> ONNXRuntimeSegmentation::m_ortSession [private]
```

Handle to the ONNXRuntime session.

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

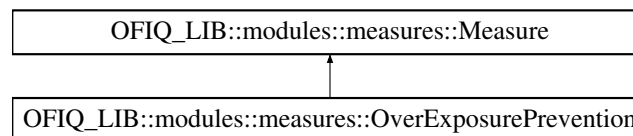
- [ONNXRTSegmentation.h](#)

## 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:



### Public Member Functions

- [OverExposurePrevention](#) (const [Configuration](#) &configuration)  
*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, [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 ) [explicit]
```

Constructor a new Over Exposure Prevention.

#### Parameters

<i>configuration</i>	Configuration object from which measure-related configuration is read.
----------------------	--

### 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

<i>session</i>	Session object computed by the <a href="#">OFIQImpl::performPreprocessing()</a> 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**

<i>in</i>	<i>faceLandmarks</i>	Landmarks to be filtered.
	<i>part</i>	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::getPairsForPart (
    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**

<i>faceLandmarks</i>	Set of face landmarks.
<i>part</i>	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:

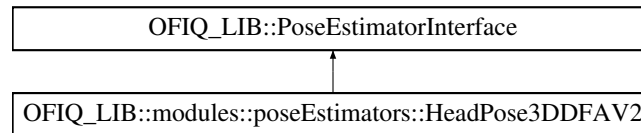
- [utils.h](#)

## 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:



### 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 [m\\_lastSessionId](#)  
*id of the session that has been used in the latest request, for internal use.*
- [EulerAngle](#) [m\\_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>
```

The first entry encodes yaw, the second encodes pitch, and the third encodes roll.



### 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

<i>session</i>	Session object containing the original facial image and pre-processing results computed by the <a href="#">OFIQImpl::performPreprocessing()</a> 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

<i>session</i>	Containing the input image for the estimation.
<i>pose</i>	Return the estimated pose.

Implemented in [OFIQ\\_LIB::modules::poseEstimators::HeadPose3DDFAV2](#).

### 7.44.5 Member Data Documentation

#### 7.44.5.1 m\_lastSessionId

```
std::string OFIQ_LIB::PoseEstimatorInterface::m_lastSessionId [private]
```

id of the session that has been used in the latest request, for internal use.

### 7.44.5.2 m\_pose

`EulerAngle OFIQ_LIB::PoseEstimatorInterface::m_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::NotInitialized](#))  
*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	<i>rawScore</i>	Computed raw score.
in	<i>scalar</i>	Computed scalar score.
in	<i>code</i>	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	<i>code</i>	The return status code; required.
in	<i>info</i>	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:

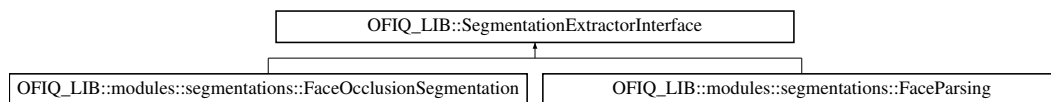
- [ofiq\\_structs.h](#)

## 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:



### 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](#) [m\\_lastSessionId](#)  
*id of the session that has been used in the latest request, for internal use.*
- [std::map](#)< [modules::segmentations::SegmentClassLabels](#), [OFIQ::Image](#) > [m\\_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 FaceOcclusionSegmentation (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

<i>session</i>	Object containing the relevant data information on the input image.
<i>faceSegment</i>	Enum of the face region that is requested.

##### Returns

OFIQ::Image& Reference 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

<i>session</i>	Object containing the relevant data information on the input image.
<i>faceSegment</i>	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 m\_lastSessionId

```
std::string OFIQ_LIB::SegmentationExtractorInterface::m_lastSessionId [private]
```

id of the session that has been used in the latest request, for internal use.

### 7.47.4.2 m\_masks

```
std::map<modules::segmentations::SegmentClassLabels, OFIQ::Image> OFIQ_LIB::Segmentation↔
ExtractorInterface::m_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  
*Access 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 & m_image`  
*Reference to the input image, connected to this session.*
- `OFIQ::FaceImageQualityAssessment & m_assessment`  
*Refernce to the FaceImageQualityAssessment object, connected to this session.*
- `std::vector< OFIQ::BoundingBox > m_detectedFaces`  
*Container for the faces found on the input image.*
- `EulerAngle m_pose`  
*Container for storing the pose information.*
- `OFIQ::FaceLandmarks m_landmarks`  
*Container for storing the landmark information.*
- `OFIQ::FaceLandmarks m_alignedFaceLandmarks`  
*Container for storing the landmark information of the aligned image.*
- `cv::Mat m_alignedFaceTransformationMatrix`  
*Container for storing the transformation matrix that led to the aligned image.*
- `cv::Mat m_alignedFace`  
*Container for storing the aligned image.*
- `cv::Mat m_alignedFacelandmarkedRegion`  
*Container for storing the landmarks of the aligned face image.*
- `cv::Mat m_faceParsingImage`  
*Container for storing the segmented face image.*
- `cv::Mat m_faceOcclusionSegmentationImage`  
*Container for storing the result of the face occlusion segmented image.*
- `std::string m_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**

<i>image</i>	Input image that shall be analysed.
<i>assessment</i>	Container to store 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]
```

Access 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>i_alignedFace</i>	
----------------------	--

**7.48.3.15 setAlignedFaceLandmarkedRegion()**

```
void OFIQ_LIB::Session::setAlignedFaceLandmarkedRegion (
    const cv::Mat & i_alignedFaceRegion )
```

Set the Aligned Face Landmarked Region.

## Parameters

<i>i_alignedFaceRegion</i>	
----------------------------	--

**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>i_landmarks</i>	
--------------------	--

**7.48.3.17 setAlignedFaceTransformationMatrix()**

```
void OFIQ_LIB::Session::setAlignedFaceTransformationMatrix (
    const cv::Mat & i_transformationMatrix )
```

Set the Aligned Face Transformation Matrix.

## Parameters

<i>i_transformationMatrix</i>	
-------------------------------	--

**7.48.3.18 setDetectedFaces()**

```
void OFIQ_LIB::Session::setDetectedFaces (
    const std::vector< OFIQ::BoundingBox > & i_boundingBoxes )
```

Set the Detected Faces.

## Parameters

<i>i_boundingBoxes</i>	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>i_segmentationImage</i>	
----------------------------	--

**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>i_parsingImage</i>	
-----------------------	--

**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>i_landmarks</i>	
--------------------	--

**7.48.3.22 setPose()**

```
void OFIQ_LIB::Session::setPose (
    const EulerAngle & i_pose )
```

Set the Pose of the input image.

## Parameters

<i>i_pose</i>	
---------------	--

## 7.48.4 Member Data Documentation

### 7.48.4.1 m\_alignedFace

```
cv::Mat OFIQ_LIB::Session::m_alignedFace [private]
```

Container for storing the aligned image.

### 7.48.4.2 m\_alignedFacelandmarkedRegion

```
cv::Mat OFIQ_LIB::Session::m_alignedFacelandmarkedRegion [private]
```

Container for storing the landmarks of the aligned face image.

### 7.48.4.3 m\_alignedFaceLandmarks

```
OFIQ::FaceLandmarks OFIQ_LIB::Session::m_alignedFaceLandmarks [private]
```

Container for storing the landmark information of the aligned image.

### 7.48.4.4 m\_alignedFaceTransformationMatrix

```
cv::Mat OFIQ_LIB::Session::m_alignedFaceTransformationMatrix [private]
```

Container for storing the transformation matrix that led to the aligned image.

### 7.48.4.5 m\_assessment

```
OFIQ::FaceImageQualityAssessment& OFIQ_LIB::Session::m_assessment [private]
```

Reference to the FaceImageQualityAssessment object, connected to this session.

### 7.48.4.6 m\_detectedFaces

```
std::vector<OFIQ::BoundingBox> OFIQ_LIB::Session::m_detectedFaces [private]
```

Container for the faces found on the input image.

#### 7.48.4.7 m\_faceOcclusionSegmentationImage

```
cv::Mat OFIQ_LIB::Session::m_faceOcclusionSegmentationImage [private]
```

Container for storing the result of the face occlusion segmented image.

#### 7.48.4.8 m\_faceParsingImage

```
cv::Mat OFIQ_LIB::Session::m_faceParsingImage [private]
```

Container for storing the segmented face image.

#### 7.48.4.9 m\_id

```
std::string OFIQ_LIB::Session::m_id [private]
```

Container for storing the id of the session.

#### 7.48.4.10 m\_image

```
const OFIQ::Image& OFIQ_LIB::Session::m_image [private]
```

Reference to the input image, connected to this session.

#### 7.48.4.11 m\_landmarks

```
OFIQ::FaceLandmarks OFIQ_LIB::Session::m_landmarks [private]
```

Container for storing the landmark information.

#### 7.48.4.12 m\_pose

```
EulerAngle OFIQ_LIB::Session::m_pose [private]
```

Container for storing the pose information.

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

- [Session.h](#)

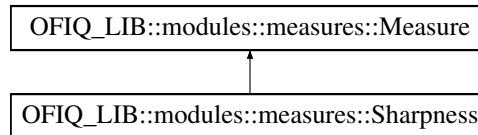


## 7.49 OFIQ\_LIB::modules::measures::Sharpness Class Reference

Implementation of the sharpness measure.

```
#include <Sharpness.h>
```

Inheritance diagram for OFIQ\_LIB::modules::measures::Sharpness:



### Public Member Functions

- [Sharpness](#) (const [Configuration](#) &configuration)  
*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, [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](#) (const [Session](#) &session, cv::Mat &faceCrop, cv::Mat &maskCrop, bool useAligned, float faceRegionAlpha) const  
*Get the cropped face region.*
- cv::Mat [GetClassifierFocusFeatures](#) (const cv::Mat &image, const cv::Mat &mask, bool applyBlur) const  
*Computation of the input features using different edge detectors.*

### Private Attributes

- `std::string m_modelFile`  
*Name of the random forest model, extracted from the configuration file.*
- `std::shared_ptr< cv::ml::RTrees > m_rtree`  
*Instance of the random forest model.*
- `bool m_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 m_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 m_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

Implementantation 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 ) [explicit]
```

Construct a new Sharpness object.

## Parameters

<i>configuration</i>	Configuration object from which measure-related configuration is read.
----------------------	--

## 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

<i>session</i>	Session object computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method.
----------------	---

Implements [OFIQ\\_LIB::modules::measures::Measure](#).

### 7.49.3.2 GetClassifierFocusFeatures()

```
cv::Mat OFIQ_LIB::modules::measures::Sharpness::GetClassifierFocusFeatures (
    const cv::Mat & image,
    const cv::Mat & mask,
    bool applyBlur ) const [private]
```

Computation of the input features using different edge detectors.

## Parameters

<i>image</i>	Input image.
<i>mask</i>	Input region of the face.
<i>applyBlur</i>	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 (
    const Session & session,
    cv::Mat & faceCrop,
    cv::Mat & maskCrop,
    bool useAligned,
    float faceRegionAlpha ) const [private]
```

Get the cropped face region.

## Parameters

<i>session</i>	Data container.
<i>faceCrop</i>	Computed crop of the face.
<i>maskCrop</i>	Mask used for the cropping. Will be computed in the method.
<i>useAligned</i>	Switch for using the aligned image.
<i>faceRegionAlpha</i>	Enlarge the face region by passing this parameter.

## 7.49.4 Member Data Documentation

### 7.49.4.1 m\_faceRegionAlpha

```
double OFIQ_LIB::modules::measures::Sharpness::m_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 m\_modelFile

```
std::string OFIQ_LIB::modules::measures::Sharpness::m_modelFile [private]
```

Name of the random forest model, extracted from the configuration file.

### 7.49.4.3 m\_numTrees

```
int OFIQ_LIB::modules::measures::Sharpness::m_numTrees [private]
```

This member stores the number of trees used for the random forest. Internal use only.

### 7.49.4.4 m\_rtree

```
std::shared_ptr<cv::ml::RTrees> OFIQ_LIB::modules::measures::Sharpness::m_rtree [private]
```

Instance of the random forest model.

### 7.49.4.5 m\_useAligned

```
bool OFIQ_LIB::modules::measures::Sharpness::m_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:

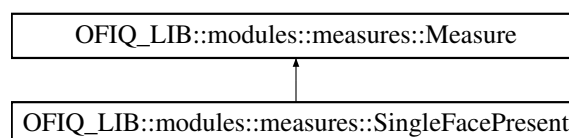
- [Measure.h](#)

## 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`:



## Public Member Functions

- [SingleFacePresent](#) (const [Configuration](#) &configuration)  
*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, [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 ) [explicit]
```

Construct a new Single Face Present object.

##### Parameters

<i>configuration</i>	Configuration object from which measure-related configuration is read.
----------------------	--

### 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

<i>session</i>	Session object containing the original facial image and pre-processing results computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method
----------------	--

Implements [OFIQ\\_LIB::modules::measures::Measure](#).

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

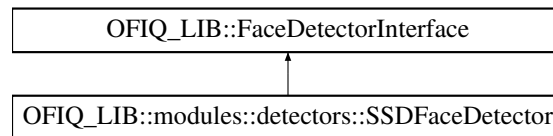
- [SingleFacePresent.h](#)

## 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:



## 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 > [m\\_dnnNet](#) {nullptr}  
*Instance of an opencv dnn::Net.*
- double [m\\_confidenceThreshold](#)  
*Confidence threshold used for the face detection. The value is read from the configuration file.*
- double [m\\_padding](#)  
*Add padding around the image (faceImage.width \* padding; faceImage.height \* padding;)*
- double [m\\_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

<i>config</i>	
---------------	--

**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

<i>session</i>	Session object computed by the <a href="#">OFIQImpl::performPreprocessing()</a> 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 m\_confidenceThreshold**

```
double OFIQ_LIB::modules::detectors::SSDFaceDetector::m_confidenceThreshold [private]
```

Confidence threshold used for the face detection. The value is read from the configuration file.

**7.52.4.2 m\_dnnNet**

```
std::shared_ptr<cv::dnn::Net> OFIQ_LIB::modules::detectors::SSDFaceDetector::m_dnnNet {nullptr}
[private]
```

Instance of an opencv dnn::Net.

### 7.52.4.3 m\_minimalRelativeFaceSize

```
double OFIQ_LIB::modules::detectors::SSDFaceDetector::m_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 m\_padding

```
double OFIQ_LIB::modules::detectors::SSDFaceDetector::m_padding [private]
```

Add padding around the image (facelImage.width \* padding; facelImage.height \* padding;)

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

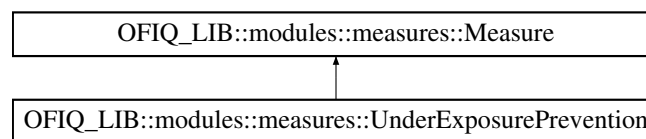
- [opencv\\_ssd\\_face\\_detector.h](#)

## 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:



### Public Member Functions

- [UnderExposurePrevention](#) (const [Configuration](#) &configuration)  
*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, [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 ) [explicit]
```

Constructor a new Under Exposure Prevention object.

#### Parameters

<i>configuration</i>	Configuration object from which measure-related configuration is read.
----------------------	--

### 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

<code>session</code>	Session object computed by the <a href="#">OFIQImpl::performPreprocessing()</a> method.
----------------------	---

Implements [OFIQ\\_LIB::modules::measures::Measure](#).

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

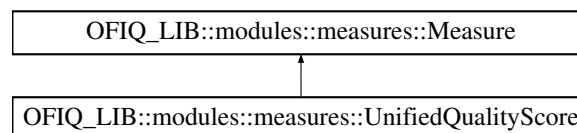
- [UnderExposurePrevention.h](#)

## 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)  
*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, [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 ) [explicit]
```

Construct a new Unified Quality Score object.

## Parameters

<i>configuration</i>	Configuration object from which measure-related configuration is read.
----------------------	--

## 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

<i>session</i>	Session object computed by the <a href="#">OFIQImpl::performPreprocessing()</a> 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, 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.

##### Author

OFIQ development team

### 8.2 mainpage.h

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

```
00001
01184 #pragma once
```

## 8.3 ofiq\_lib.h File Reference

Class describing the interface to the OFIQ.

```
#include <stdint>
#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, 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.

#### 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         OFIQ_EXPORT void getVersion(int& major, int& minor, int& patch) const;
00136
00137     };
00138 }
00139
00140 #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, 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.

#### 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 dummyAssessment;
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
00134         std::unique_ptr<OFIQ_LIB::modules::measures::Executor> CreateExecutor();
00135
00136
00141         void CreateNetworks();
00142
00150         void performPreprocessing(Session& session);
00151
00159         void alignFaceImage(Session& session) const;
00160     };
00161 }
00162
00163 #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 <stdint>
#include <cstring>
#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 describing 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::CropOfTheFacelImage` = -0x54 , `OFIQ::LeftwardCropOfTheFacelImage` = 0x54 , `OFIQ::RightwardCropOfTheFacelImage` = 0x55 ,  
`OFIQ::MarginAboveOfTheFacelImage` = 0x56 ,  
`OFIQ::MarginBelowOfTheFacelImage` = 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, 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.

#### 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 <stdint>
00031 #include <cstring>
00032 #include <iostream>
00033 #include <map>
00034 #include <memory>
00035 #include <string>
00036 #include <utility>
00037 #include <vector>
00038
00042 namespace OFIQ
00043 {
00048     struct Image
00049     {
00050
00052         uint16_t width{ 0 };
00054         uint16_t height{ 0 };
00056         uint8_t depth{ 24 };
00061         std::shared_ptr<uint8_t> data;
00062
00066         Image() = default;
00067
00081         Image(uint16_t width, uint16_t height, uint8_t depth, const std::shared_ptr<uint8_t>& data)
00082             : width{ width },
00083               height{ height },
00084               depth{ depth },
00085               data{ data }
00086         {
00087         }
00088
00090         size_t size() const { return (static_cast<size_t>(width) * height * (depth / 8)); }
00091
00104         void deepcopy(uint16_t width, uint16_t height, uint8_t depth, const std::shared_ptr<uint8_t>&
data)

```

```

00105     {
00106         this->width = width;
00107         this->height = height;
00108         this->depth = depth;
00109         size_t size = this->size();
00110         this->data.reset(new uint8_t[size], std::default_delete<uint8_t[]>());
00111         memcpy(this->data.get(), data.get(), size);
00112     }
00113 };
00114
00115
00120 enum class ReturnCode
00121 {
00122     Success = 0,
00123     ImageReadingError,
00124     ImageWritingError,
00125     MissingConfigParamError,
00126     UnknownConfigParamError,
00127     FaceDetectionError,
00128     FaceLandmarkExtractionError,
00129     FaceOcclusionSegmentationError,
00130     FaceParsingError,
00131     UnknownError,
00132     QualityAssessmentError,
00133     NotImplemented
00134 };
00135
00140 inline std::ostream& operator<<(std::ostream& s, const ReturnCode& rc)
00141 {
00142     switch (rc)
00143     {
00144     case ReturnCode::Success:
00145         return (s << "Success");
00146     case ReturnCode::UnknownError:
00147         return (s << "Unknown Error");
00148     case ReturnCode::QualityAssessmentError:
00149         return (s << "Failure to generate a quality score on the input image");
00150     case ReturnCode::NotImplemented:
00151         return (s << "Function is not implemented");
00152     default:
00153         return (s << "Undefined error");
00154     }
00155 }
00156
00160 struct ReturnStatus
00161 {
00162     ReturnCode code{ ReturnCode::UnknownError };
00163     std::string info;
00164
00165     ReturnStatus() = default;
00166
00170     ReturnStatus(const ReturnCode code, const std::string& info = "")
00171         : code{code},
00172         info{info}
00173     {
00174     }
00175 };
00176
00180 enum class QualityMeasure
00181 {
00182     UnifiedQualityScore = 0x41,
00183     BackgroundUniformity = 0x42,
00184     IlluminationUniformity = 0x43,
00185     Luminance = 0x44,
00186     LuminanceMean = 0x44,
00187     LuminanceVariance = 0x45,
00188     UnderExposurePrevention = 0x46,
00189     OverExposurePrevention = 0x47,
00190     DynamicRange = 0x48,
00191     Sharpness = 0x49,
00192     CompressionArtifacts = 0x4a,
00193     NaturalColour = 0x4b,
00194     SingleFacePresent = 0x4c,
00195     EyesOpen = 0x4d,
00196     MouthClosed = 0x4e,
00197     EyesVisible = 0x4f,
00198     MouthOcclusionPrevention = 0x50,
00199     FaceOcclusionPrevention = 0x51,
00200     InterEyeDistance = 0x52,
00201     HeadSize = 0x53,
00202     CropOfTheFaceImage = 0x54,
00203     LeftwardCropOfTheFaceImage = 0x54,
00204     RightwardCropOfTheFaceImage = 0x55,
00205     MarginAboveOfTheFaceImage = 0x56,
00206     MarginBelowOfTheFaceImage = 0x57,
00207     HeadPose = 0x58,
00208     HeadPoseYaw = 0x58,

```



```

00266         HeadPosePitch = 0x59,
00268         HeadPoseRoll = 0x5a,
00270         ExpressionNeutrality = 0x5b,
00272         NoHeadCoverings = 0x5c,
00274         NotSet = -1
00275     };
00276
00281     enum class QualityMeasureReturnCode
00282     {
00284         Success = 0,
00286         FailureToAssess,
00288         NotInitialized
00289     };
00290
00295     struct QualityMeasureResult
00296     {
00298         double rawScore{ -1 };
00302         double scalar{ -1 };
00304         QualityMeasureReturnCode code{ QualityMeasureReturnCode::NotInitialized };
00305
00310         QualityMeasureResult() = default;
00311
00319         QualityMeasureResult(double rawScore, double scalar = -1, QualityMeasureReturnCode code =
QualityMeasureReturnCode::NotInitialized)
00320             : rawScore{rawScore},
00321               scalar{scalar},
00322               code{code}
00323         {
00324         }
00325     };
00326
00332     using QualityAssessments = std::map<QualityMeasure, QualityMeasureResult>;
00333
00338     enum class FaceDetectorType
00339     {
00341         OPENCVSSD,
00343         NotSet
00344     };
00345
00352     struct BoundingBox
00353     {
00356         int16_t xleft{ -1 };
00359         int16_t ytop{ -1 };
00361         int16_t width{ -1 };
00363         int16_t height{ -1 };
00364
00366         FaceDetectorType faceDetector = FaceDetectorType::NotSet;
00367
00372         BoundingBox() = default;
00373
00383         BoundingBox(int16_t xleft, int16_t ytop, int16_t width, int16_t height, FaceDetectorType
i_faceDetector)
00384             : xleft{xleft},
00385               ytop{ytop},
00386               width{width},
00387               height{height},
00388               faceDetector{i_faceDetector}
00389         {
00390         }
00391     };
00392
00397     struct LandmarkPoint
00398     {
00403         int16_t x{ -1 };
00408         int16_t y{ -1 };
00409
00414         LandmarkPoint() = default;
00415
00422         LandmarkPoint(int16_t i_x, int16_t i_y)
00423             : x{i_x},
00424               y{i_y}
00425         {
00426         }
00427     };
00428
00433     using Landmarks = std::vector<LandmarkPoint>;
00434
00439     enum class LandmarkType
00440     {
00442         LM_98,
00444         NotSet
00445     };
00446
00447
00448
00453     struct FaceLandmarks

```

```

00454     {
00455         LandmarkType type{ LandmarkType::NotSet };
00456         Landmarks landmarks;
00457
00458         FaceLandmarks() = default;
00459     };
00460
00461     struct FaceImageQualityAssessment
00462     {
00463
00464         QualityAssessments qAssessments;
00465
00466         BoundingBox boundingBox;
00467
00468         FaceImageQualityAssessment() = default;
00469
00470         FaceImageQualityAssessment(
00471             const QualityAssessments& qAssessments, const BoundingBox& boundingBox)
00472             : qAssessments{qAssessments},
00473               boundingBox{boundingBox}
00474         {
00475         }
00476     };
00477 }
00478
00479 #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, 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.

#### 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, 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.

#### 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.

**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, 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.

**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> m_dnnNet{nullptr};
00076
00081         double m_confidenceThreshold;
00082
00087         double m_padding;
00088
00093         double m_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](#) {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22}  
*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, 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.

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
00086     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, 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.

#### 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     private:
00074
00078         std::unique_ptr<ADNetFaceLandmarkExtractorImpl> landmarkExtractor_;
00079     };
00080 }
```

## 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, 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.

#### 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

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.

#### 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::FOREHEAD  
 }

*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, 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.

#### 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, 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.

#### 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
00058     protected:
00065         virtual OFIQ::FaceLandmarks updateLandmarks(OFIQ_LIB::Session& session) = 0;
00066     };
00067 }
```

## 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, 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.

## 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},
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,
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, 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.

#### 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

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.

**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:
00050         explicit BackgroundUniformity(
00051             const Configuration& configuration);
00052
00060         void Execute(OFIQ_LIB::Session & session) override;
00061
00062     private:
00067         uint16_t m_targetHeight = 292;
00068
00073         uint16_t m_targetWidth = 354;
00074
00080         uint16_t m_cropLeft = 62;
00081
00087         uint16_t m_cropRight = 62;
00088
00094         uint16_t m_cropTop = 0;
00095
00101         uint16_t m_cropBottom = 210;
00102
00109         uint16_t m_erosionKernelSize = 4;
00110     };
00111 }
```

**8.33 CompressionArtifacts.h File Reference**

Provides a class implementing 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 implementing 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, 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.

#### 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:
00068         explicit CompressionArtifacts(const Configuration& configuration);
00069
00078         void Execute(OFIQ_LIB::Session& session) override;
00079
00080     private:
00087         int m_crop;
00088
00096         int m_dim;
00097
00101         ONNXRuntimeSegmentation m_onnxRuntimeEnv;
00102     };
00103 }
```

## 8.35 CropOfTheFacelImage.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::CropOfTheFacelImage](#)  
*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, 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.

#### Author

OFIQ development 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:
00049         explicit CropOfTheFaceImage(const Configuration& configuration);
00050
00057         void Execute(OFIQ_LIB::Session & session) override;
00058     };
00059 }
```

## 8.37 DynamicRange.h File Reference

Provides a class implementing 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 implementing 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, 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.

### 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:
00050         explicit DynamicRange(
00051             const Configuration& configuration);
00052
00058         void Execute(OFIQ_LIB::Session & session) override;
00059     };
00060 }
```

## 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::ExecutorLogActive](#) = 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

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.

#### 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 ExecutorLogActive = 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             : m_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 m_measures; }
00076
00077     private:
00082         std::vector<std::unique_ptr<Measure>> m_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, 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.

#### 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:
00054         explicit ExpressionNeutrality(
00055             const Configuration& configuration);
00056
00062         void Execute(OFIQ_LIB::Session& session) override;
00063
00064     private:
00069         ONNXRuntimeSegmentation m_onnxRuntimeEnvCNN1;
00070
00075         ONNXRuntimeSegmentation m_onnxRuntimeEnvCNN2;
00076
00081         std::shared_ptr<cv::ml::Boost> m_classifier;
00082     };
00083 }
```

## 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

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.

**Author**

OFIQ development team

**8.44 EyesOpen.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 EyesOpen : public Measure
00044     {
00045     public:
00050         explicit EyesOpen(const Configuration& configuration);
00051
00060         void Execute(OFIQ_LIB::Session & session) override;
00061     };
00062 }
```

## 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.

#### 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, 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.

#### 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:
00050         explicit EyesVisible(const Configuration& configuration);
00051
00061         void Execute(OFIQ_LIB::Session & session) override;
00062     };
00063 }
```

## 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, 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.

## 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:
00051         explicit FaceOcclusionPrevention(
00052             const Configuration& configuration);
00053
00064         void Execute(OFIQ_LIB::Session & session) override;
00065     };
00066 }

```

## 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, 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.

#### 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:
00047         explicit HeadPose(
00048             const Configuration& configuration);
00049
00057         void Execute(OFIQ_LIB::Session & session) override;
00058     };
00059 }
```

## 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.

#### 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, 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.

#### 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:
00049         explicit HeadSize(
00050             const Configuration& configuration);
00051
00057         void Execute(OFIQ_LIB::Session & session) override;
00058     };
00059 }
```

## 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, 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.

## Author

OFIQ development team

## 8.54 IlluminationUniformity.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 {
00044     class IlluminationUniformity : public Measure
00045     {
00046     public:
00051         explicit IlluminationUniformity(const Configuration& configuration);
00052
00061         void Execute(OFIQ_LIB::Session & session) override;
00062     };
00063 }
```

## 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, 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.

#### 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:
00050         explicit InterEyeDistance(const Configuration& configuration);
00051
00060         void Execute(OFIQ_LIB::Session & session) override;
00061     };
00062 }
```

## 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.

#### 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, 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.

#### 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:
00049         explicit Luminance(const Configuration& configuration);
00050
00058         void Execute(OFIQ_LIB::Session & session) override;
00059     };
00060 }
```

## 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, 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.

## 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                 OFIQ::QualityMeasure measure)
00148             : configuration{configuration}, m_measure(measure)
00149         {
00150         }
00151
00161         virtual void Execute(OFIQ_LIB::Session& session) = 0;
00162
00166         virtual ~Measure() = default;
00167
00175         virtual std::string GetName() const;
00176
00181         virtual OFIQ::QualityMeasure GetQualityMeasure() const;
00182
00196         void SetQualityMeasure(OFIQ_LIB::Session& session, OFIQ::QualityMeasure measure, double
rawValue, OFIQ::QualityMeasureReturnCode code);
00197
00198     protected:
00206         static double Sigmoid(double x, double x0, double w)
00207         {
00208             return 1.0 / (1 + exp((x0 - x) / w));
00209         }
00210
00223         void AddSigmoid(OFIQ::QualityMeasure measure, const SigmoidParameters& defaultValues);
00224
00237         void AddSigmoid(const std::string& key, SigmoidParameters defaultValues);
00238
00246         double ExecuteScalarConversion(OFIQ::QualityMeasure measure, double rawValue);
00247
00255         double ExecuteScalarConversion(const std::string& key, double rawValue);
00256
00261         const Configuration& configuration;
00262
00263     private:
00274         static double ScalarConversion(double rawValue, const SigmoidParameters& par)
00275         {
00276             double scalarScore = par.h * (par.a + par.s * Sigmoid(rawValue, par.x0, par.w));
00277             if (par.round)
00278                 scalarScore = round(scalarScore);
00279             if (scalarScore < 0.0)
00280             {
00281                 scalarScore = 0.0;
00282             }
00283             else if (scalarScore > 100.0)
00284             {
00285                 scalarScore = 100.0;
00286             }
00287             return scalarScore;
00288         }
00289
00294         std::map<std::string, SigmoidParameters, std::less<> m_sigmoidMap;
00295
00301         static std::string GetMeasureName(OFIQ::QualityMeasure measure);
00302
00310         static std::string ExpandKey(std::string_view rawKey);
00311
00317         OFIQ::QualityMeasure m_measure = OFIQ::QualityMeasure::NotSet;
00318     };
00319 }

```

## 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, 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.

#### 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
00059         static std::unique_ptr<Measure> CreateMeasure(
00060             const OFIQ::QualityMeasure measure,
00061             const Configuration& configuration);
00062     };
00063 }
```



## 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, 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.

#### 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:
00049         explicit MouthClosed(const Configuration& configuration);
00050
00059         void Execute(OFIQ_LIB::Session& session) override;
00060     };
00061 }
```

## 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

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.

### 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:
00049         explicit MouthOcclusionPrevention(const Configuration& configuration);
00050
00062         void Execute(OFIQ_LIB::Session & session) override;
00063     };
00064 }
```

## 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, 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.

#### Author

OFIQ development team

## 8.68 NaturalColour.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 NaturalColour : public Measure
00044     {
00045     public:
00050         explicit NaturalColour(const Configuration& configuration);
00051
00060         void Execute(OFIQ_LIB::Session & session) override;
00061
00062     private:
00069         cv::Mat CreateMaskedImage(const OFIQ::FaceLandmarks& landmarks, const cv::Mat& cvImage) const;
00070
00081         cv::Mat ReduceImageToRegionOfInterest(
00082             const cv::Mat& maskedImage,
00083             const cv::Rect& leftRegionOfInterest,
00084             const cv::Rect& rightRegionOfInterest) const;
00085
00099         double CalculateScore(double meanChannelA, double meanChannelB) const;
00100     };
00101 }
```

## 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, 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.

#### 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:
00062         explicit NoHeadCoverings(const Configuration& configuration);
00063
00080         void Execute(OFIQ_LIB::Session & session) override;
00081
00082     private:
00089         double m_t0;
00090
00097         double m_tl;
00098
00105         double m_w;
00106
00113         double m_x0;
00114     };
00115 }
```

## 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

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.

### 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:
00049         explicit OverExposurePrevention(const Configuration& configuration);
00050
00056         void Execute(OFIQ_LIB::Session & session) override;
00057     };
00058 }
```

## 8.73 Sharpness.h File Reference

Provides a class implementing the sharpness measure.

```
#include "Measure.h"
```

### Classes

- class [OFIQ\\_LIB::modules::measures::Sharpness](#)  
*Implementantation 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, 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.

#### 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:
00050         explicit Sharpness(const Configuration& configuration);
00051
00057         void Execute(OFIQ_LIB::Session & session) override;
00058
00059     private:
00060
00064         std::string m_modelFile;
00065
00070         std::shared_ptr<cv::ml::RTrees> m_rtree;
00071
00077         bool m_useAligned;
00078
```



```

00084         double m_faceRegionAlpha;
00085
00090         int m_numTrees;
00091
00101         void GetCroppedImages(
00102             const Session& session,
00103             cv::Mat& faceCrop,
00104             cv::Mat& maskCrop,
00105             bool useAligned,
00106             float faceRegionAlpha) const;
00107
00116         cv::Mat GetClassifierFocusFeatures(const cv::Mat& image, const cv::Mat& mask, bool applyBlur)
00117             const;
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, 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.

#### 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:
00048         explicit SingleFacePresent(const Configuration& configuration);
00049
00056         void Execute(OFIQ_LIB::Session & session) override;
00057     };
00058 }
```

## 8.77 UnderExposurePrevention.h File Reference

Provides a class implementing 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 implementing the under-exposure 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, 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.

**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:
00048         explicit UnderExposurePrevention(const Configuration& configuration);
00049
00055         void Execute(OFIQ_LIB::Session & session) override;
00056     };
00057 }
```

## 8.79 UnifiedQualityScore.h File Reference

Provides a class implementing 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 implementing 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, 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.

#### 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 <ONNXRuntimeSegmentation.h>
00033
00037 namespace OFIQ_LIB::modules::measures
00038 {
00046     class UnifiedQualityScore : public Measure
00047     {
00048     public:
00053         explicit UnifiedQualityScore(const Configuration& configuration);
00054
00064         void Execute(OFIQ_LIB::Session & session) override;
00065
00066     private:
00071         ONNXRuntimeSegmentation m_onnxRuntimeEnv;
00072     };
00073 }
```

## 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, 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.

@briefPP provides 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/3DDFA\\_V2](https://github.com/cleardusk/3DDFA_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/3DDFA\\_V2](https://github.com/cleardusk/3DDFA_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, 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.

#### 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 m_paramPoseEstimatorModel;
00075
00079         Ort::Env m_ortenv;
00080
00084         std::unique_ptr<Ort::Session> m_ortSession;
00085
00089         int64_t m_expectedImageWidth = 0;
00090
00094         int64_t m_expectedImageHeight = 0;
00095
00099         int64_t m_expectedImageNumberOfChannels = 0;
00100
00104         int64_t m_numberOfInputElements = 0;
00105
00109         std::array<int64_t, 4> m_inputShape;
00110
00118         cv::Mat CropImage(const cv::Mat& image, const OFIQ::BoundingBox& biggestFace) const;
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, 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.

#### 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:
00049         using EulerAngle = std::array<double, 3>;
00050
00055         virtual ~PoseEstimatorInterface() = default;
00056
00064         EulerAngle& estimatePose(OFIQ_LIB::Session& session);
00065
00066     protected:
00073         virtual void updatePose(OFIQ_LIB::Session& session, EulerAngle& pose) = 0;
00074
00075     private:
00080         std::string m_lastSessionId;
00081
00086         EulerAngle m_pose;
00087     };
00088 }
```



## 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, 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.

#### 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);
00070         ~FaceOcclusionSegmentation() override = default;
00071
00072     protected:
00073         OFIQ::Image UpdateMask(
00093             OFIQ_LIB::Session& session, modules::segmentations::SegmentClassLabels faceSegment)
00094         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> m_segmentationImage;
00118
00123         const std::string m_modelConfigItem = "params.measures.FaceOcclusionSegmentation.model_path";
00124
00128         const int m_cropLeft = 96;
00129
00133         const int m_cropRight = 96;
00134
00138         const int m_cropTop = 96;
00139
00143         const int m_cropBottom = 96;
00144
00150         const int m_scaledWidth = 224;
00151
00157         const int m_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, 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.

#### 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     protected:
00100         OFIQ::Image UpdateMask(
00126             OFIQ_LIB::Session& session, modules::segmentations::SegmentClassLabels faceSegment)
00127     override;
00128
00129     private:
00130
00134         ONNXRuntimeSegmentation m_onnxRuntimeEnv;
00135
00141         std::shared_ptr<cv::Mat> m_segmentationImage;
00142
00148         const std::string m_modelConfigItem = "params.measures.FaceParsing.model_path";
00149
00153         const int m_imageSize = 400;
00154
00158         const int m_cropLeft = 30;
00159
00163         const int m_cropRight = 30;
00164
00168         const int m_cropBottom = 60;
00169
00178         static cv::Mat CreateBlob(const cv::Mat& image, int i_imageSize_one_dim);
00179
00191         static std::shared_ptr<cv::Mat> CalculateClassIds(
00192             const cv::Mat& resultImage,
00193             int i_imageSize_one_dim);
00194
00195         /**
00196          * @brief Derives the private member \link segmentationImage\endlink
00197          * from the facial image data provided by the session object.
00198          * @details Implements CNN processing step of \link
00199 OFIQ_LIB::modules::segmentations::FaceParsing::UpdateMask()
00199          * UpdateMask()\endlink.
00200          * @param session Session object containing the original facial image and pre-processing
00201 results
00201          * computed by the \link OFIQ_LIB::OFIQImpl::performPreprocessing()
00202          * OFIQImpl::performPreprocessing()\endlink method.
00203          */
00204         void SetImage(const OFIQ_LIB::Session& session);
00205     };
00206 }

```

## 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.

**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, 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.

**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_memoryInfo = Ort::MemoryInfo::CreateCpu(OrtDeviceAllocator, OrtMemTypeCPU);
00054
00059     std::array<int64_t, 4> m_inputShape;
00060
00065     std::unique_ptr<Ort::Session> m_ortSession;
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() const;
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::segmentations::r\\_brow](#) ,  
[OFIQ\\_LIB::modules::segmentations::l\\_eye](#) , [OFIQ\\_LIB::modules::segmentations::r\\_eye](#) , [OFIQ\\_LIB::modules::segmentations::l\\_ear](#) ,  
[OFIQ\\_LIB::modules::segmentations::r\\_ear](#) , [OFIQ\\_LIB::modules::segmentations::ear\\_r](#) , [OFIQ\\_LIB::modules::segmentations::mouth](#) ,  
[OFIQ\\_LIB::modules::segmentations::u\\_lip](#) , [OFIQ\\_LIB::modules::segmentations::l\\_lip](#) , [OFIQ\\_LIB::modules::segmentations::neck\\_l](#) ,  
[OFIQ\\_LIB::modules::segmentations::neck\\_r](#) , [OFIQ\\_LIB::modules::segmentations::cloth](#) , [OFIQ\\_LIB::modules::segmentations::hair](#) , [OFIQ\\_LIB::modules::segmentations::face](#) }  
*Enum class of the different face regions 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, 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.

#### 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 m_lastSessionId; };
00174
00175     private:
00180         std::string m_lastSessionId;
00185         std::map<modules::segmentations::SegmentClassLabels, OFIQ::Image> m_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, 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.

#### 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_view 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.*
- [OFIQ\\_EXPORT OFIQ::ReturnStatus OFIQ\\_LIB::readImageFromByteArray](#) (const std::vector< unsigned char > &buffer, [OFIQ::Image](#) &image)  
*Read image from byte array.*

### 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

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.

#### Author

OFIQ development team

## 8.98 image\_io.h

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

```
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
00054     OFIQ_EXPORT OFIQ::ReturnStatus
00055         readImageFromByteArray(const std::vector<unsigned char>& buffer, OFIQ::Image& image);
00056
00057 }
00058
00059 #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** &rightEyeCenter, 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, 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.

### 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,
00118         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
00090     std::shared_ptr<PoseEstimatorInterface> poseEstimator;
00091
00097     std::shared_ptr<SegmentationExtractorInterface> faceOcclusionExtractor;
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, 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.

**Author**

OFIQ development team

**8.104 OFIQError.h**[Go to the documentation of this file.](#)

```

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 m_extendedMessage.c_str(); }
00057
00063         OFIQ::ReturnCode whatCode() const noexcept { return m_returnCode; }
00064
00065     private:
00070         OFIQ::ReturnCode m_returnCode;
00071
00076         std::string m_message;
00077
00082         std::string m_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

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.

#### 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             : m_image{image},
00063               m_assessment{assessment},
00064               m_id{GenerateId()}
00065         {
00066         }
00067
00072         const OFIQ::Image& image() const { return m_image; }
00073
00078         OFIQ::FaceImageQualityAssessment& assessment() { return m_assessment; }
00079
00085         const std::string& Id() const { return m_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
```

```

00108     void setPose(const EulerAngle& i_pose);
00109
00115     EulerAngle getPose() const;
00116
00122     void setLandmarks(const OFIQ::FaceLandmarks& i_landmarks);
00123
00129     OFIQ::FaceLandmarks getLandmarks() const;
00130
00131
00137     void setAlignedFaceLandmarks(const OFIQ::FaceLandmarks& i_landmarks);
00138
00144     OFIQ::FaceLandmarks getAlignedFaceLandmarks() const;
00145
00151     void setAlignedFaceTransformationMatrix(const cv::Mat & i_transformationMatrix);
00152
00153
00159     cv::Mat getAlignedFaceTransformationMatrix() const;
00160
00161
00167     void setAlignedFace(const cv::Mat & i_alignedFace);
00168
00174     cv::Mat getAlignedFace() const;
00175
00181     void setAlignedFaceLandmarkedRegion(const cv::Mat & i_alignedFaceRegion);
00182
00188     cv::Mat getAlignedFaceLandmarkedRegion() const;
00189
00195     void setFaceParsingImage(const cv::Mat& i_parsingImage);
00196
00202     cv::Mat getFaceParsingImage() const;
00203
00209     void setFaceOcclusionSegmentationImage(const cv::Mat& i_segmentationImage);
00210
00216     cv::Mat getFaceOcclusionSegmentationImage() const;
00217
00218 private:
00223     const OFIQ::Image& m_image;
00224
00229     OFIQ::FaceImageQualityAssessment& m_assessment;
00234     std::vector<OFIQ::BoundingBox> m_detectedFaces;
00235
00240     EulerAngle m_pose;
00241
00246     OFIQ::FaceLandmarks m_landmarks;
00247
00252     OFIQ::FaceLandmarks m_alignedFaceLandmarks;
00253
00258     cv::Mat m_alignedFaceTransformationMatrix;
00259
00264     cv::Mat m_alignedFace;
00265
00270     cv::Mat m_alignedFaceLandmarkedRegion;
00271
00276     cv::Mat m_faceParsingImage;
00277
00282     cv::Mat m_faceOcclusionSegmentationImage;
00283
00289     std::string GenerateId() const;
00290
00295     std::string m_id;
00296 };
00297 }

```

## 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.*

### 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, 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.

## 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
00131     OFIQ_EXPORT void calculateEyeCenter(
00132         const OFIQ::FaceLandmarks& faceLandmarks,
00133         Point2f& leftEyeCenter,
00134         Point2f& rightEyeCenter);
00135
00136     OFIQ_EXPORT OFIQ::Image MakeGreyImage(uint16_t width, uint16_t height);
00137
00138     OFIQ_EXPORT float tmetric(const OFIQ::FaceLandmarks& faceLandmarks);
00139 }
00140 #endif
```



# Index

- ~ADNetFaceLandmarkExtractor
  - OFIQ\_LIB::modules::landmarks::ADNetFaceLandmarkExtractor, [68](#)
- ~FaceDetectorInterface
  - OFIQ\_LIB::FaceDetectorInterface, [97](#)
- ~FaceLandmarkExtractorInterface
  - OFIQ\_LIB::FaceLandmarkExtractorInterface, [100](#)
- ~FaceOcclusionSegmentation
  - OFIQ\_LIB::modules::segmentations::FaceOcclusionSegmentation, [110](#)
- ~FaceParsing
  - OFIQ\_LIB::modules::segmentations::FaceParsing, [114](#)
- ~HeadPose3DDFAV2
  - OFIQ\_LIB::modules::poseEstimators::HeadPose3DDFAV2, [121](#)
- ~Interface
  - OFIQ::Interface, [133](#)
- ~Measure
  - OFIQ\_LIB::modules::measures::Measure, [142](#)
- ~OFIQImpl
  - OFIQ\_LIB::OFIQImpl, [164](#)
- ~ONNXRuntimeSegmentation
  - ONNXRuntimeSegmentation, [169](#)
- ~PoseEstimatorInterface
  - OFIQ\_LIB::PoseEstimatorInterface, [177](#)
- ~SSDFaceDetector
  - OFIQ\_LIB::modules::detectors::SSDFaceDetector, [203](#)
- ~SegmentationExtractorInterface
  - OFIQ\_LIB::SegmentationExtractorInterface, [182](#)
- a
  - OFIQ\_LIB::modules::measures::SigmoidParameters, [198](#)
- AddSigmoid
  - OFIQ\_LIB::modules::measures::Measure, [142](#)
- adnet\_FaceMap.h, [221](#), [223](#)
- adnet\_landmarks.h, [224](#), [225](#)
- ADNetFaceLandmarkExtractor
  - OFIQ\_LIB::modules::landmarks::ADNetFaceLandmarkExtractor, [68](#)
- alignFacelImage
  - OFIQ\_LIB::OFIQImpl, [165](#)
- alignImage
  - OFIQ\_LIB, [47](#)
- AllDetectors.h, [218](#), [219](#)
- AllLandmarks.h, [225](#), [226](#)
- AllMeasures.h, [232](#), [234](#)
- AllPoseEstimators.h, [269](#)
- assessment
  - OFIQ\_LIB::Session, [186](#)
- background
  - OFIQ\_LIB::modules::segmentations, [65](#)
- BackgroundUniformity
  - OFIQ, [43](#)
  - OFIQ\_LIB::modules::measures::BackgroundUniformity, [71](#)
  - BackgroundUniformity.h, [234](#), [235](#)
- BoundingBox
  - OFIQ::BoundingBox, [73](#)
- boundingBox
  - OFIQ::FaceImageQualityAssessment, [99](#)
- CalculateClassIds
  - OFIQ\_LIB::modules::segmentations::FaceParsing, [115](#)
- CalculateExposure
  - OFIQ\_LIB, [48](#)
- calculateEyeCenter
  - OFIQ\_LIB, [48](#)
- CalculateReferencePoints
  - OFIQ\_LIB, [48](#)
- CalculateRegionOfInterest
  - OFIQ\_LIB, [49](#)
- CalculateScore
  - OFIQ\_LIB::modules::measures::NaturalColour, [154](#)
- CHIN
  - OFIQ\_LIB::modules::landmarks, [58](#)
- chin
  - OFIQ\_LIB::modules::landmarks::adnet, [59](#)
- cloth
  - OFIQ\_LIB::modules::segmentations, [66](#)
- code
  - OFIQ::QualityMeasureResult, [179](#)
  - OFIQ::ReturnStatus, [180](#)
- ColorConvert
  - OFIQ\_LIB, [49](#)
- CompressionArtifacts
  - OFIQ, [44](#)
  - OFIQ\_LIB::modules::measures::CompressionArtifacts, [76](#)
- CompressionArtifacts.h, [235](#), [236](#)
- ComputeBrightnessAspect
  - OFIQ\_LIB, [50](#)
- config
  - OFIQ\_LIB::OFIQImpl, [167](#)
- Configuration

- OFIQ\_LIB::Configuration, 79
- configuration
  - OFIQ\_LIB::modules::measures::Measure, 146
- Configuration.h, 279, 280
- contour
  - OFIQ\_LIB::modules::landmarks::adnet, 59
- ConvertBGRToCIELAB
  - OFIQ\_LIB, 50
- copyToCvImage
  - OFIQ\_LIB, 50
- CreateBlob
  - OFIQ\_LIB::modules::segmentations::FaceParsing, 115
- CreateExecutor
  - OFIQ\_LIB::OFIQImpl, 165
- CreateMaskedImage
  - OFIQ\_LIB::modules::measures::NaturalColour, 155
- CreateMeasure
  - OFIQ\_LIB::modules::measures::MeasureFactory, 147
- CreateNetworks
  - OFIQ\_LIB::OFIQImpl, 165
- CropImage
  - OFIQ\_LIB::modules::poseEstimators::HeadPose3DDFAV2, 121
- CropOfTheFacelImage
  - OFIQ, 44
  - OFIQ\_LIB::modules::measures::CropOfTheFacelImage, 84
- CropOfTheFacelImage.h, 237, 238
- Cubic
  - OFIQ\_LIB, 51
- cv, 41
- data
  - OFIQ::Image, 129
- deepcopy
  - OFIQ::Image, 129
- depth
  - OFIQ::Image, 129
- detectFaces
  - OFIQ\_LIB::FaceDetectorInterface, 97
- detectors.h, 219, 220
- dummyAssement
  - OFIQ\_LIB::OFIQImpl, 167
- dummyImage
  - OFIQ\_LIB::OFIQImpl, 167
- DynamicRange
  - OFIQ, 44
  - OFIQ\_LIB::modules::measures::DynamicRange, 86
- DynamicRange.h, 238, 239
- ear\_r
  - OFIQ\_LIB::modules::segmentations, 66
- estimatePose
  - OFIQ\_LIB::PoseEstimatorInterface, 177
- EulerAngle
  - OFIQ\_LIB, 47
- OFIQ\_LIB::PoseEstimatorInterface, 176
- Execute
  - OFIQ\_LIB::modules::measures::BackgroundUniformity, 71
  - OFIQ\_LIB::modules::measures::CompressionArtifacts, 77
  - OFIQ\_LIB::modules::measures::CropOfTheFacelImage, 85
  - OFIQ\_LIB::modules::measures::DynamicRange, 87
  - OFIQ\_LIB::modules::measures::ExpressionNeutrality, 91
  - OFIQ\_LIB::modules::measures::EyesOpen, 93
  - OFIQ\_LIB::modules::measures::EyesVisible, 96
  - OFIQ\_LIB::modules::measures::FaceOcclusionPrevention, 107
  - OFIQ\_LIB::modules::measures::HeadPose, 119
  - OFIQ\_LIB::modules::measures::HeadSize, 125
  - OFIQ\_LIB::modules::measures::IlluminationUniformity, 127
  - OFIQ\_LIB::modules::measures::InterEyeDistance, 132
  - OFIQ\_LIB::modules::measures::Luminance, 139
  - OFIQ\_LIB::modules::measures::Measure, 142
  - OFIQ\_LIB::modules::measures::MouthClosed, 150
  - OFIQ\_LIB::modules::measures::MouthOcclusionPrevention, 152
  - OFIQ\_LIB::modules::measures::NaturalColour, 155
  - OFIQ\_LIB::modules::measures::NoHeadCoverings, 160
  - OFIQ\_LIB::modules::measures::OverExposurePrevention, 173
  - OFIQ\_LIB::modules::measures::Sharpness, 195
  - OFIQ\_LIB::modules::measures::SingleFacePresent, 201
  - OFIQ\_LIB::modules::measures::UnderExposurePrevention, 206
  - OFIQ\_LIB::modules::measures::UnifiedQualityScore, 208
- ExecuteAll
  - OFIQ\_LIB::modules::measures::Executor, 88
- ExecuteScalarConversion
  - OFIQ\_LIB::modules::measures::Measure, 143
- Executor
  - OFIQ\_LIB::modules::measures::Executor, 88
- Executor.h, 239, 241
- ExecutorLogActive
  - OFIQ\_LIB::modules::measures, 64
- ExpandKey
  - OFIQ\_LIB::modules::measures::Measure, 144
- ExposureRange
  - OFIQ\_LIB, 47
- ExpressionNeutrality
  - OFIQ, 44
  - OFIQ\_LIB::modules::measures::ExpressionNeutrality, 90

- ExpressionNeutrality.h, [241](#), [242](#)
- extractLandmarks
  - OFIQ\_LIB::FaceLandmarkExtractorInterface, [100](#)
- eye\_g
  - OFIQ\_LIB::modules::segmentations, [66](#)
- EyesOpen
  - OFIQ, [44](#)
  - OFIQ\_LIB::modules::measures::EyesOpen, [93](#)
- EyesOpen.h, [242](#), [243](#)
- EyesVisible
  - OFIQ, [44](#)
  - OFIQ\_LIB::modules::measures::EyesVisible, [95](#)
- EyesVisible.h, [244](#), [245](#)
- face
  - OFIQ\_LIB::modules::segmentations, [66](#)
- FACE\_CONTOUR
  - OFIQ\_LIB::modules::landmarks, [58](#)
- FaceDetectionError
  - OFIQ, [45](#)
- faceDetector
  - OFIQ::BoundingBox, [74](#)
  - OFIQ\_LIB::NeuronalNetworkContainer, [157](#)
- FaceDetectorType
  - OFIQ, [43](#)
- FaceImageQualityAssessment
  - OFIQ::FaceImageQualityAssessment, [98](#)
- FaceLandmarkExtractionError
  - OFIQ, [45](#)
- FaceLandmarks
  - OFIQ::FaceLandmarks, [101](#)
- FaceMap
  - OFIQ\_LIB::modules::landmarks, [57](#)
  - OFIQ\_LIB::modules::landmarks::adnet, [60](#)
- FaceMeasures
  - OFIQ\_LIB::modules::landmarks::FaceMeasures, [102](#)
- FaceMeasures.h, [226](#), [227](#)
- faceOcclusionExtractor
  - OFIQ\_LIB::NeuronalNetworkContainer, [157](#)
- FaceOcclusionPrevention
  - OFIQ, [44](#)
  - OFIQ\_LIB::modules::measures::FaceOcclusionPrevention, [107](#)
- FaceOcclusionPrevention.h, [245](#), [246](#)
- FaceOcclusionSegmentation
  - OFIQ\_LIB::modules::segmentations::FaceOcclusionSegmentation, [109](#)
- FaceOcclusionSegmentation.h, [273](#), [274](#)
- FaceOcclusionSegmentationError
  - OFIQ, [45](#)
- FacePairMap
  - OFIQ\_LIB::modules::landmarks, [57](#)
  - OFIQ\_LIB::modules::landmarks::adnet, [60](#)
- FaceParsing
  - OFIQ\_LIB::modules::segmentations::FaceParsing, [114](#)
- FaceParsing.h, [274](#), [275](#)
- FaceParsingError
  - OFIQ, [45](#)
- FaceParts
  - OFIQ\_LIB::modules::landmarks, [58](#)
- FaceParts.h, [228](#), [229](#)
- FailureToAssess
  - OFIQ, [44](#)
- findLargestBoundingBox
  - OFIQ\_LIB, [51](#)
- FOREHEAD
  - OFIQ\_LIB::modules::landmarks, [58](#)
- forehead
  - OFIQ\_LIB::modules::landmarks::adnet, [60](#)
- GenerateId
  - OFIQ\_LIB::Session, [186](#)
- getAlignedFace
  - OFIQ\_LIB::Session, [186](#)
- getAlignedFaceLandmarkedRegion
  - OFIQ\_LIB::Session, [186](#)
- getAlignedFaceLandmarks
  - OFIQ\_LIB::Session, [186](#)
- getAlignedFaceTransformationMatrix
  - OFIQ\_LIB::Session, [187](#)
- GetBool
  - OFIQ\_LIB::Configuration, [79](#)
- GetClassifierFocusFeatures
  - OFIQ\_LIB::modules::measures::Sharpness, [195](#)
- GetCroppedImages
  - OFIQ\_LIB::modules::measures::Sharpness, [195](#)
- getDataDir
  - OFIQ\_LIB::Configuration, [80](#)
- getDetectedFaces
  - OFIQ\_LIB::Session, [187](#)
- GetDistance
  - OFIQ\_LIB::modules::landmarks::FaceMeasures, [102](#), [103](#)
- GetFaceMask
  - OFIQ\_LIB::modules::landmarks::FaceMeasures, [103](#)
- GetFaceOcclusionSegmentation
  - OFIQ\_LIB::modules::segmentations::FaceOcclusionSegmentation, [110](#)
- getFaceOcclusionSegmentationImage
  - OFIQ\_LIB::Session, [187](#)
- getFaceParsingImage
  - OFIQ\_LIB::Session, [187](#)
- getFacePart
  - OFIQ\_LIB::modules::landmarks::PartExtractor, [173](#)
- getImplementation
  - OFIQ::Interface, [133](#)
- getLandmarks
  - OFIQ\_LIB::Session, [187](#)
- GetLastSessionId
  - OFIQ\_LIB::SegmentationExtractorInterface, [182](#)
- GetLuminanceImageFromBGR
  - OFIQ\_LIB, [52](#)
- GetMask
  - OFIQ\_LIB::SegmentationExtractorInterface, [182](#)

- GetMaxPairDistance
  - OFIQ\_LIB::modules::landmarks::FaceMeasures, 103
- GetMeasureName
  - OFIQ\_LIB::modules::measures::Measure, 144
- GetMeasures
  - OFIQ\_LIB::modules::measures::Executor, 88
- GetMiddle
  - OFIQ\_LIB::modules::landmarks::FaceMeasures, 104, 105
- GetName
  - OFIQ\_LIB::modules::measures::Measure, 144
- GetNormalizedHistogram
  - OFIQ\_LIB, 52
- GetNumber
  - OFIQ\_LIB::Configuration, 80
- getNumberOfOutputNodes
  - ONNXRuntimeSegmentation, 169
- getPairsForPart
  - OFIQ\_LIB::modules::landmarks::PartExtractor, 174
- getPose
  - OFIQ\_LIB::Session, 188
- GetQualityMeasure
  - OFIQ\_LIB::modules::measures::Measure, 145
- GetString
  - OFIQ\_LIB::Configuration, 81
- GetStringList
  - OFIQ\_LIB::Configuration, 82
- getVersion
  - OFIQ::Interface, 133
- h
  - OFIQ\_LIB::modules::measures::SigmoidParameters, 198
- hair
  - OFIQ\_LIB::modules::segmentations, 66
- hat
  - OFIQ\_LIB::modules::segmentations, 66
- HeadPose
  - OFIQ, 44
  - OFIQ\_LIB::modules::measures::HeadPose, 119
- HeadPose.h, 246, 247
- HeadPose3DDFAV2
  - OFIQ\_LIB::modules::poseEstimators::HeadPose3DDFAV2, 121
- HeadPose3DDFAV2.h, 269, 271
- HeadPosePitch
  - OFIQ, 44
- HeadPoseRoll
  - OFIQ, 44
- HeadPoseYaw
  - OFIQ, 44
- HeadSize
  - OFIQ, 44
  - OFIQ\_LIB::modules::measures::HeadSize, 125
- HeadSize.h, 248, 249
- height
  - OFIQ::BoundingBox, 74
- OFIQ::Image, 130
- Id
  - OFIQ\_LIB::Session, 188
- IlluminationUniformity
  - OFIQ, 43
  - OFIQ\_LIB::modules::measures::IlluminationUniformity, 127
- IlluminationUniformity.h, 249, 250
- Image
  - OFIQ::Image, 128
- image
  - OFIQ\_LIB::Session, 188
- image\_io.h, 281, 282
- image\_utils.h, 282, 283
- ImageReadingError
  - OFIQ, 45
- ImageWritingError
  - OFIQ, 45
- info
  - OFIQ::ReturnStatus, 180
- init\_session
  - ONNXRuntimeSegmentation, 169
- initialize
  - OFIQ::Interface, 133
  - OFIQ\_LIB::OFIImpl, 165
  - ONNXRuntimeSegmentation, 170
- InterEyeDistance
  - OFIQ, 44
  - OFIQ\_LIB::modules::landmarks::FaceMeasures, 105
  - OFIQ\_LIB::modules::measures::InterEyeDistance, 131
- InterEyeDistance.h, 250, 251
- I\_brow
  - OFIQ\_LIB::modules::segmentations, 65
- I\_ear
  - OFIQ\_LIB::modules::segmentations, 66
- I\_eye
  - OFIQ\_LIB::modules::segmentations, 65
- I\_lip
  - OFIQ\_LIB::modules::segmentations, 66
- landmarkExtractor
  - OFIQ\_LIB::NeuronalNetworkContainer, 157
- landmarkExtractor\_
  - OFIQ\_LIB::modules::landmarks::ADNetFaceLandmarkExtractor, 69
- LandmarkId
  - OFIQ\_LIB::modules::landmarks, 57
- LandmarkIdPair
  - OFIQ\_LIB::modules::landmarks, 58
- LandmarkIdPairs
  - OFIQ\_LIB::modules::landmarks, 58
- LandmarkIds
  - OFIQ\_LIB::modules::landmarks, 58
- LandmarkPair
  - OFIQ\_LIB::modules::landmarks::LandmarkPair, 135



- LandmarkPoint
  - OFIQ::LandmarkPoint, [137](#)
- Landmarks
  - OFIQ, [42](#)
- landmarks
  - OFIQ::FaceLandmarks, [101](#)
- landmarks.h, [230](#), [231](#)
- LandmarkType
  - OFIQ, [43](#)
- LEFT\_EYE
  - OFIQ\_LIB::modules::landmarks, [58](#)
- LEFT\_EYE\_CORNERS
  - OFIQ\_LIB::modules::landmarks, [58](#)
- leftEye
  - OFIQ\_LIB::modules::landmarks::adnet, [60](#)
- leftEyeCorners
  - OFIQ\_LIB::modules::landmarks::adnet, [60](#)
- LeftwardCropOfTheFacelImage
  - OFIQ, [44](#)
- LM\_98
  - OFIQ, [43](#)
- log
  - OFIQ\_LIB::modules::measures, [64](#)
- Lower
  - OFIQ\_LIB::modules::landmarks::LandmarkPair, [136](#)
- Luminance
  - OFIQ, [43](#)
  - OFIQ\_LIB::modules::measures::Luminance, [139](#)
- Luminance.h, [252](#), [253](#)
- LuminanceMean
  - OFIQ, [43](#)
- LuminanceVariance
  - OFIQ, [43](#)
- m\_alignedFace
  - OFIQ\_LIB::Session, [191](#)
- m\_alignedFacelandmarkedRegion
  - OFIQ\_LIB::Session, [191](#)
- m\_alignedFaceLandmarks
  - OFIQ\_LIB::Session, [191](#)
- m\_alignedFaceTransformationMatrix
  - OFIQ\_LIB::Session, [191](#)
- m\_assessment
  - OFIQ\_LIB::Session, [191](#)
- m\_classifier
  - OFIQ\_LIB::modules::measures::ExpressionNeutrality, [91](#)
- m\_confidenceThreshold
  - OFIQ\_LIB::modules::detectors::SSDFaceDetector, [203](#)
- m\_crop
  - OFIQ\_LIB::modules::measures::CompressionArtifacts, [77](#)
- m\_cropBottom
  - OFIQ\_LIB::modules::measures::BackgroundUniformity, [71](#)
  - OFIQ\_LIB::modules::segmentations::FaceOcclusionSegmentation, [111](#)
- OFIQ\_LIB::modules::segmentations::FaceParsing, [116](#)
- m\_cropLeft
  - OFIQ\_LIB::modules::measures::BackgroundUniformity, [71](#)
  - OFIQ\_LIB::modules::segmentations::FaceOcclusionSegmentation, [111](#)
  - OFIQ\_LIB::modules::segmentations::FaceParsing, [116](#)
- m\_cropRight
  - OFIQ\_LIB::modules::measures::BackgroundUniformity, [72](#)
  - OFIQ\_LIB::modules::segmentations::FaceOcclusionSegmentation, [111](#)
  - OFIQ\_LIB::modules::segmentations::FaceParsing, [116](#)
- m\_cropTop
  - OFIQ\_LIB::modules::measures::BackgroundUniformity, [72](#)
  - OFIQ\_LIB::modules::segmentations::FaceOcclusionSegmentation, [111](#)
- m\_dataDir
  - OFIQ\_LIB::Configuration, [82](#)
- m\_detectedFaces
  - OFIQ\_LIB::Session, [191](#)
- m\_dim
  - OFIQ\_LIB::modules::measures::CompressionArtifacts, [77](#)
- m\_dnnNet
  - OFIQ\_LIB::modules::detectors::SSDFaceDetector, [203](#)
- m\_emptySession
  - OFIQ\_LIB::OFIQImpl, [167](#)
- m\_erosionKernelSize
  - OFIQ\_LIB::modules::measures::BackgroundUniformity, [72](#)
- m\_executorPtr
  - OFIQ\_LIB::OFIQImpl, [167](#)
- m\_expectedImageHeight
  - OFIQ\_LIB::modules::poseEstimators::HeadPose3DDFAV2, [122](#)
- m\_expectedImageNumberOfChannels
  - OFIQ\_LIB::modules::poseEstimators::HeadPose3DDFAV2, [122](#)
- m\_expectedImageWidth
  - OFIQ\_LIB::modules::poseEstimators::HeadPose3DDFAV2, [122](#)
- m\_extendedMessage
  - OFIQ\_LIB::OFIQError, [163](#)
- m\_faceOcclusionSegmentationImage
  - OFIQ\_LIB::Session, [191](#)
- m\_faceParsingImage
  - OFIQ\_LIB::Session, [192](#)
- m\_faceRegionAlpha
  - OFIQ\_LIB::modules::measures::Sharpness, [196](#)
- m\_id
  - OFIQ\_LIB::Session, [192](#)
- m\_image

- OFIQ\_LIB::Session, 192
- m\_imageSize
  - OFIQ\_LIB::modules::segmentations::FaceParsing, 117
- m\_inputShape
  - OFIQ\_LIB::modules::poseEstimators::HeadPose3DDFAV2, 122
  - ONNXRuntimeSegmentation, 170
- m\_landmarks
  - OFIQ\_LIB::Session, 192
- m\_lastSessionId
  - OFIQ\_LIB::PoseEstimatorInterface, 177
  - OFIQ\_LIB::SegmentationExtractorInterface, 183
- m\_masks
  - OFIQ\_LIB::SegmentationExtractorInterface, 183
- m\_measure
  - OFIQ\_LIB::modules::measures::Measure, 146
- m\_measures
  - OFIQ\_LIB::modules::measures::Executor, 88
- m\_memoryInfo
  - ONNXRuntimeSegmentation, 170
- m\_message
  - OFIQ\_LIB::OFIQError, 163
- m\_minimalRelativeFaceSize
  - OFIQ\_LIB::modules::detectors::SSDFaceDetector, 203
- m\_modelConfigItem
  - OFIQ\_LIB::modules::segmentations::FaceOcclusionSegmentation, 111
  - OFIQ\_LIB::modules::segmentations::FaceParsing, 117
- m\_modelFile
  - OFIQ\_LIB::modules::measures::Sharpness, 196
- m\_numberOfInputElements
  - OFIQ\_LIB::modules::poseEstimators::HeadPose3DDFAV2, 123
- m\_numTrees
  - OFIQ\_LIB::modules::measures::Sharpness, 196
- m\_onnxRuntimeEnv
  - OFIQ\_LIB::modules::measures::CompressionArtifacts, 77
  - OFIQ\_LIB::modules::measures::UnifiedQualityScore, 208
  - OFIQ\_LIB::modules::segmentations::FaceOcclusionSegmentation, 111
  - OFIQ\_LIB::modules::segmentations::FaceParsing, 117
- m\_onnxRuntimeEnvCNN1
  - OFIQ\_LIB::modules::measures::ExpressionNeutrality, 91
- m\_onnxRuntimeEnvCNN2
  - OFIQ\_LIB::modules::measures::ExpressionNeutrality, 91
- m\_ortenv
  - OFIQ\_LIB::modules::poseEstimators::HeadPose3DDFAV2, 123
  - ONNXRuntimeSegmentation, 170
- m\_ortSession
  - OFIQ\_LIB::modules::poseEstimators::HeadPose3DDFAV2, 123
  - ONNXRuntimeSegmentation, 170
- m\_padding
  - OFIQ\_LIB::modules::detectors::SSDFaceDetector, 204
- m\_paramPoseEstimatorModel
  - OFIQ\_LIB::modules::poseEstimators::HeadPose3DDFAV2, 123
- m\_pose
  - OFIQ\_LIB::PoseEstimatorInterface, 177
  - OFIQ\_LIB::Session, 192
- m\_returnCode
  - OFIQ\_LIB::OFIQError, 163
- m\_rtree
  - OFIQ\_LIB::modules::measures::Sharpness, 196
- m\_scaledHeight
  - OFIQ\_LIB::modules::segmentations::FaceOcclusionSegmentation, 112
- m\_scaledWidth
  - OFIQ\_LIB::modules::segmentations::FaceOcclusionSegmentation, 112
- m\_segmentationImage
  - OFIQ\_LIB::modules::segmentations::FaceOcclusionSegmentation, 112
  - OFIQ\_LIB::modules::segmentations::FaceParsing, 117
- m\_signalMap
  - OFIQ\_LIB::modules::measures::Measure, 146
- m\_t0
  - OFIQ\_LIB::modules::measures::NoHeadCoverings, 160
- m\_t1
  - OFIQ\_LIB::modules::measures::NoHeadCoverings, 160
- m\_targetHeight
  - OFIQ\_LIB::modules::measures::BackgroundUniformity, 72
- m\_targetWidth
  - OFIQ\_LIB::modules::measures::BackgroundUniformity, 72
- m\_useAligned
  - OFIQ\_LIB::modules::measures::Sharpness, 196
- m\_weight
  - OFIQ\_LIB::modules::measures::NoHeadCoverings, 160
- m\_x0
  - OFIQ\_LIB::modules::measures::NoHeadCoverings, 161
- mainpage.h, 209
- MakeGreyImage
  - OFIQ\_LIB, 52
- makeSquareBoundingBox
  - OFIQ\_LIB, 53
- makeSquareBoundingBoxWithPadding
  - OFIQ\_LIB, 53
- MarginAboveOfTheFaceImage
  - OFIQ, 44

- MarginBelowOfTheFacelImage
  - OFIQ, [44](#)
- Measure
  - OFIQ\_LIB::modules::measures::Measure, [141](#)
- Measure.h, [253](#), [254](#)
- MeasureFactory
  - OFIQ\_LIB::modules::measures::MeasureFactory, [147](#)
- MeasureFactory.h, [255](#), [256](#)
- MissingConfigParamError
  - OFIQ, [45](#)
- mouth
  - OFIQ\_LIB::modules::segmentations, [66](#)
- MOUTH\_CENTER
  - OFIQ\_LIB::modules::landmarks, [58](#)
- MOUTH\_INNER
  - OFIQ\_LIB::modules::landmarks, [58](#)
- MOUTH\_OUTER
  - OFIQ\_LIB::modules::landmarks, [58](#)
- MouthClosed
  - OFIQ, [44](#)
  - OFIQ\_LIB::modules::measures::MouthClosed, [149](#)
- MouthClosed.h, [257](#), [258](#)
- mouthInner
  - OFIQ\_LIB::modules::landmarks::adnet, [61](#)
- MouthOcclusionPrevention
  - OFIQ, [44](#)
  - OFIQ\_LIB::modules::measures::MouthOcclusionPrevention, [151](#)
- MouthOcclusionPrevention.h, [258](#), [259](#)
- mouthOuter
  - OFIQ\_LIB::modules::landmarks::adnet, [61](#)
- NaturalColour
  - OFIQ, [44](#)
  - OFIQ\_LIB::modules::measures::NaturalColour, [154](#)
- NaturalColour.h, [259](#), [260](#)
- neck
  - OFIQ\_LIB::modules::segmentations, [66](#)
- neck\_l
  - OFIQ\_LIB::modules::segmentations, [66](#)
- networks
  - OFIQ\_LIB::OFIQImpl, [167](#)
- NeuronalNetworkContainer
  - OFIQ\_LIB::NeuronalNetworkContainer, [156](#)
- NeuronalNetworkContainer.h, [284](#)
- NoHeadCoverings
  - OFIQ, [44](#)
  - OFIQ\_LIB::modules::measures::NoHeadCoverings, [159](#)
- NoHeadCoverings.h, [261](#), [262](#)
- nose
  - OFIQ\_LIB::modules::segmentations, [66](#)
- NOSETIP
  - OFIQ\_LIB::modules::landmarks, [58](#)
- nosetip
  - OFIQ\_LIB::modules::landmarks::adnet, [61](#)
- NotImplemented
  - OFIQ, [45](#)
- NotInitialized
  - OFIQ, [44](#)
- NotSet
  - OFIQ, [43](#), [44](#)
- OFIQ, [41](#)
  - BackgroundUniformity, [43](#)
  - CompressionArtifacts, [44](#)
  - CropOfTheFacelImage, [44](#)
  - DynamicRange, [44](#)
  - ExpressionNeutrality, [44](#)
  - EyesOpen, [44](#)
  - EyesVisible, [44](#)
  - FaceDetectionError, [45](#)
  - FaceDetectorType, [43](#)
  - FaceLandmarkExtractionError, [45](#)
  - FaceOcclusionPrevention, [44](#)
  - FaceOcclusionSegmentationError, [45](#)
  - FaceParsingError, [45](#)
  - FailureToAssess, [44](#)
  - HeadPose, [44](#)
  - HeadPosePitch, [44](#)
  - HeadPoseRoll, [44](#)
  - HeadPoseYaw, [44](#)
  - HeadSize, [44](#)
  - IlluminationUniformity, [43](#)
  - ImageReadingError, [45](#)
  - ImageWritingError, [45](#)
  - InterEyeDistance, [44](#)
  - Landmarks, [42](#)
  - LandmarkType, [43](#)
  - LeftwardCropOfTheFacelImage, [44](#)
  - LM\_98, [43](#)
  - Luminance, [43](#)
  - LuminanceMean, [43](#)
  - LuminanceVariance, [43](#)
  - MarginAboveOfTheFacelImage, [44](#)
  - MarginBelowOfTheFacelImage, [44](#)
  - MissingConfigParamError, [45](#)
  - MouthClosed, [44](#)
  - MouthOcclusionPrevention, [44](#)
  - NaturalColour, [44](#)
  - NoHeadCoverings, [44](#)
  - NotImplemented, [45](#)
  - NotInitialized, [44](#)
  - NotSet, [43](#), [44](#)
  - OPENCVSSD, [43](#)
  - operator<<, [45](#)
  - OverExposurePrevention, [43](#)
  - QualityAssessmentError, [45](#)
  - QualityAssessments, [42](#)
  - QualityMeasure, [43](#)
  - QualityMeasureReturnCode, [44](#)
  - ReturnCode, [44](#)
  - RightwardCropOfTheFacelImage, [44](#)
  - Sharpness, [44](#)
  - SingleFacePresent, [44](#)
  - Success, [44](#), [45](#)

- UnderExposurePrevention, 43
- UnifiedQualityScore, 43
- UnknownConfigParamError, 45
- UnknownError, 45
- OFIQ::BoundingBox, 73
  - BoundingBox, 73
  - faceDetector, 74
  - height, 74
  - width, 74
  - xleft, 74
  - ytop, 74
- OFIQ::FacelImageQualityAssessment, 98
  - boundingBox, 99
  - FacelImageQualityAssessment, 98
  - qAssessments, 99
- OFIQ::FaceLandmarks, 101
  - FaceLandmarks, 101
  - landmarks, 101
  - type, 101
- OFIQ::Image, 128
  - data, 129
  - deepcopy, 129
  - depth, 129
  - height, 130
  - Image, 128
  - size, 129
  - width, 130
- OFIQ::Interface, 132
  - ~Interface, 133
  - getImplementation, 133
  - getVersion, 133
  - initialize, 133
  - scalarQuality, 134
  - vectorQuality, 134
- OFIQ::LandmarkPoint, 136
  - LandmarkPoint, 137
  - x, 137
  - y, 137
- OFIQ::QualityMeasureResult, 178
  - code, 179
  - QualityMeasureResult, 178
  - rawScore, 179
  - scalar, 179
- OFIQ::ReturnStatus, 179
  - code, 180
  - info, 180
  - ReturnStatus, 180
- OFIQ\_EXPORT
  - ofiq\_lib.h, 211
- OFIQ\_LIB, 45
  - alignImage, 47
  - CalculateExposure, 48
  - calculateEyeCenter, 48
  - CalculateReferencePoints, 48
  - CalculateRegionOfInterest, 49
  - ColorConvert, 49
  - ComputeBrightnessAspect, 50
  - ConvertBGRTToCIELAB, 50
  - copyToCvImage, 50
  - Cubic, 51
  - EulerAngle, 47
  - ExposureRange, 47
  - findLargestBoundingBox, 51
  - GetLuminanceImageFromBGR, 52
  - GetNormalizedHistogram, 52
  - MakeGreyImage, 52
  - makeSquareBoundingBox, 53
  - makeSquareBoundingBoxWithPadding, 53
  - readImage, 53
  - readImageFromByteArray, 55
  - tmetric, 55
- ofiq\_lib.h, 210, 211
  - OFIQ\_EXPORT, 211
- OFIQ\_LIB::Configuration, 78
  - Configuration, 79
  - GetBool, 79
  - getDataDir, 80
  - GetNumber, 80
  - GetString, 81
  - GetStringList, 82
  - m\_dataDir, 82
  - parameters, 82
  - SetDataDir, 82
- OFIQ\_LIB::FaceDetectorInterface, 96
  - ~FaceDetectorInterface, 97
  - detectFaces, 97
  - UpdateFaces, 97
- OFIQ\_LIB::FaceLandmarkExtractorInterface, 99
  - ~FaceLandmarkExtractorInterface, 100
  - extractLandmarks, 100
  - updateLandmarks, 100
- OFIQ\_LIB::modules, 55
- OFIQ\_LIB::modules::detectors, 56
- OFIQ\_LIB::modules::detectors::SSDFaceDetector, 201
  - ~SSDFaceDetector, 203
  - m\_confidenceThreshold, 203
  - m\_dnnNet, 203
  - m\_minimalRelativeFaceSize, 203
  - m\_padding, 204
  - SSDFaceDetector, 202
  - UpdateFaces, 203
- OFIQ\_LIB::modules::landmarks, 56
  - CHIN, 58
  - FACE\_CONTOUR, 58
  - FaceMap, 57
  - FacePairMap, 57
  - FaceParts, 58
  - FOREHEAD, 58
  - LandmarkId, 57
  - LandmarkIdPair, 58
  - LandmarkIdPairs, 58
  - LandmarkIds, 58
  - LEFT\_EYE, 58
  - LEFT\_EYE\_CORNERS, 58
  - MOUTH\_CENTER, 58
  - MOUTH\_INNER, 58

- MOUTH\_OUTER, 58
- NOSETIP, 58
- RIGHT\_EYE, 58
- RIGHT\_EYE\_CORNERS, 58
- OFIQ\_LIB::modules::landmarks::adnet, 59
  - chin, 59
  - contour, 59
  - FaceMap, 60
  - FacePairMap, 60
  - forehead, 60
  - leftEye, 60
  - leftEyeCorners, 60
  - mouthInner, 61
  - mouthOuter, 61
  - nosetip, 61
  - pairsInnerLip, 61
  - pairsLeftEye, 61
  - pairsMouthCenter, 61
  - pairsRightEye, 62
  - rightEye, 62
  - rightEyeCorners, 62
- OFIQ\_LIB::modules::landmarks::ADNetFaceLandmarkExtractor, 67
  - ~ADNetFaceLandmarkExtractor, 68
  - ADNetFaceLandmarkExtractor, 68
  - landmarkExtractor\_, 69
  - updateLandmarks, 68
- OFIQ\_LIB::modules::landmarks::FaceMeasures, 102
  - FaceMeasures, 102
  - GetDistance, 102, 103
  - GetFaceMask, 103
  - GetMaxPairDistance, 103
  - GetMiddle, 104, 105
  - InterEyeDistance, 105
- OFIQ\_LIB::modules::landmarks::LandmarkPair, 135
  - LandmarkPair, 135
  - Lower, 136
  - Upper, 136
- OFIQ\_LIB::modules::landmarks::PartExtractor, 173
  - getFacePart, 173
  - getPairsForPart, 174
- OFIQ\_LIB::modules::measures, 62
  - ExecutorLogActive, 64
  - log, 64
- OFIQ\_LIB::modules::measures::BackgroundUniformity, 69
  - BackgroundUniformity, 71
  - Execute, 71
  - m\_cropBottom, 71
  - m\_cropLeft, 71
  - m\_cropRight, 72
  - m\_cropTop, 72
  - m\_erosionKernelSize, 72
  - m\_targetHeight, 72
  - m\_targetWidth, 72
- OFIQ\_LIB::modules::measures::CompressionArtifacts, 75
  - CompressionArtifacts, 76
- Execute, 77
  - m\_crop, 77
  - m\_dim, 77
  - m\_onnxRuntimeEnv, 77
- OFIQ\_LIB::modules::measures::CropOfTheFacelImage, 83
  - CropOfTheFacelImage, 84
  - Execute, 85
- OFIQ\_LIB::modules::measures::DynamicRange, 85
  - DynamicRange, 86
  - Execute, 87
- OFIQ\_LIB::modules::measures::Executor, 87
  - ExecuteAll, 88
  - Executor, 88
  - GetMeasures, 88
  - m\_measures, 88
- OFIQ\_LIB::modules::measures::ExpressionNeutrality, 89
  - Execute, 91
  - ExpressionNeutrality, 90
  - m\_classifier, 91
- m\_onnxRuntimeEnvCNN1, 91
- m\_onnxRuntimeEnvCNN2, 91
- OFIQ\_LIB::modules::measures::EyesOpen, 92
  - Execute, 93
  - EyesOpen, 93
- OFIQ\_LIB::modules::measures::EyesVisible, 94
  - Execute, 96
  - EyesVisible, 95
- OFIQ\_LIB::modules::measures::FaceOcclusionPrevention, 106
  - Execute, 107
  - FaceOcclusionPrevention, 107
- OFIQ\_LIB::modules::measures::HeadPose, 117
  - Execute, 119
  - HeadPose, 119
- OFIQ\_LIB::modules::measures::HeadSize, 123
  - Execute, 125
  - HeadSize, 125
- OFIQ\_LIB::modules::measures::IlluminationUniformity, 125
  - Execute, 127
  - IlluminationUniformity, 127
- OFIQ\_LIB::modules::measures::InterEyeDistance, 130
  - Execute, 132
  - InterEyeDistance, 131
- OFIQ\_LIB::modules::measures::Luminance, 138
  - Execute, 139
  - Luminance, 139
- OFIQ\_LIB::modules::measures::Measure, 140
  - ~Measure, 142
  - AddSigmoid, 142
  - configuration, 146
  - Execute, 142
  - ExecuteScalarConversion, 143
  - ExpandKey, 144
  - GetMeasureName, 144
  - GetName, 144

- GetQualityMeasure, 145
- m\_measure, 146
- m\_sigmoidMap, 146
- Measure, 141
- ScalarConversion, 145
- SetQualityMeasure, 145
- Sigmoid, 146
- OFIQ\_LIB::modules::measures::MeasureFactory, 147
  - CreateMeasure, 147
  - MeasureFactory, 147
- OFIQ\_LIB::modules::measures::MouthClosed, 148
  - Execute, 150
  - MouthClosed, 149
- OFIQ\_LIB::modules::measures::MouthOcclusionPrevention, 150
  - Execute, 152
  - MouthOcclusionPrevention, 151
- OFIQ\_LIB::modules::measures::NaturalColour, 152
  - CalculateScore, 154
  - CreateMaskedImage, 155
  - Execute, 155
  - NaturalColour, 154
  - ReduceImageToRegionOfInterest, 155
- OFIQ\_LIB::modules::measures::NoHeadCoverings, 158
  - Execute, 160
  - m\_t0, 160
  - m\_t1, 160
  - m\_w, 160
  - m\_x0, 161
  - NoHeadCoverings, 159
- OFIQ\_LIB::modules::measures::OverExposurePrevention, 171
  - Execute, 173
  - OverExposurePrevention, 172
- OFIQ\_LIB::modules::measures::Sharpness, 193
  - Execute, 195
  - GetClassifierFocusFeatures, 195
  - GetCroppedImages, 195
  - m\_faceRegionAlpha, 196
  - m\_modelFile, 196
  - m\_numTrees, 196
  - m\_rtree, 196
  - m\_useAligned, 196
  - Sharpness, 194
- OFIQ\_LIB::modules::measures::SigmoidParameters, 197
  - a, 198
  - h, 198
  - Reset, 198
  - round, 198
  - s, 199
  - setInverse, 198
  - SigmoidParameters, 198
  - w, 199
  - x0, 199
- OFIQ\_LIB::modules::measures::SingleFacePresent, 199
  - Execute, 201
  - SingleFacePresent, 201
- OFIQ\_LIB::modules::measures::UnderExposurePrevention, 204
  - Execute, 206
  - UnderExposurePrevention, 205
- OFIQ\_LIB::modules::measures::UnifiedQualityScore, 206
  - Execute, 208
  - m\_onnxRuntimeEnv, 208
  - UnifiedQualityScore, 207
- OFIQ\_LIB::modules::poseEstimators, 64
- OFIQ\_LIB::modules::poseEstimators::HeadPose3DDFAV2, 119
  - ~HeadPose3DDFAV2, 121
  - CropImage, 121
  - HeadPose3DDFAV2, 121
  - m\_expectedImageHeight, 122
  - m\_expectedImageNumberOfChannels, 122
  - m\_expectedImageWidth, 122
  - m\_inputShape, 122
  - m\_numberOfInputElements, 123
  - m\_ortenv, 123
  - m\_ortSession, 123
  - m\_paramPoseEstimatorModel, 123
  - updatePose, 122
- OFIQ\_LIB::modules::segmentations, 65
  - background, 65
  - cloth, 66
  - ear\_r, 66
  - eye\_g, 66
  - face, 66
  - hair, 66
  - hat, 66
  - l\_brow, 65
  - l\_ear, 66
  - l\_eye, 65
  - l\_lip, 66
  - mouth, 66
  - neck, 66
  - neck\_l, 66
  - nose, 66
  - r\_brow, 65
  - r\_ear, 66
  - r\_eye, 65
  - SegmentClassLabels, 65
  - skin, 65
  - u\_lip, 66
- OFIQ\_LIB::modules::segmentations::FaceOcclusionSegmentation, 108
  - ~FaceOcclusionSegmentation, 110
  - FaceOcclusionSegmentation, 109
  - GetFaceOcclusionSegmentation, 110
  - m\_cropBottom, 111
  - m\_cropLeft, 111
  - m\_cropRight, 111
  - m\_cropTop, 111
  - m\_modelConfigItem, 111
  - m\_onnxRuntimeEnv, 111

- [m\\_scaledHeight](#), 112
  - [m\\_scaledWidth](#), 112
  - [m\\_segmentationImage](#), 112
  - [UpdateMask](#), 110
- [OFIQ\\_LIB::modules::segmentations::FaceParsing](#), 112
  - [~FaceParsing](#), 114
  - [CalculateClassIds](#), 115
  - [CreateBlob](#), 115
  - [FaceParsing](#), 114
  - [m\\_cropBottom](#), 116
  - [m\\_cropLeft](#), 116
  - [m\\_cropRight](#), 116
  - [m\\_imageSize](#), 117
  - [m\\_modelConfigItem](#), 117
  - [m\\_onnxRuntimeEnv](#), 117
  - [m\\_segmentationImage](#), 117
  - [SetImage](#), 115
  - [UpdateMask](#), 116
- [OFIQ\\_LIB::NeuralNetworkContainer](#), 156
  - [faceDetector](#), 157
  - [faceOcclusionExtractor](#), 157
  - [landmarkExtractor](#), 157
  - [NeuralNetworkContainer](#), 156
  - [poseEstimator](#), 157
  - [segmentationExtractor](#), 157
- [OFIQ\\_LIB::OFIQError](#), 161
  - [m\\_extendedMessage](#), 163
  - [m\\_message](#), 163
  - [m\\_returnCode](#), 163
  - [OFIQError](#), 162
  - [what](#), 162
  - [whatCode](#), 162
- [OFIQ\\_LIB::OFIQImpl](#), 163
  - [~OFIQImpl](#), 164
  - [alignFacelImage](#), 165
  - [config](#), 167
  - [CreateExecutor](#), 165
  - [CreateNetworks](#), 165
  - [dummyAssessment](#), 167
  - [dummyImage](#), 167
  - [initialize](#), 165
  - [m\\_emptySession](#), 167
  - [m\\_executorPtr](#), 167
  - [networks](#), 167
  - [OFIQImpl](#), 164
  - [performPreprocessing](#), 166
  - [scalarQuality](#), 166
  - [vectorQuality](#), 166
- [OFIQ\\_LIB::Point2i](#), 175
  - [x](#), 175
  - [y](#), 175
- [OFIQ\\_LIB::PoseEstimatorInterface](#), 176
  - [~PoseEstimatorInterface](#), 177
  - [estimatePose](#), 177
  - [EulerAngle](#), 176
  - [m\\_lastSessionId](#), 177
  - [m\\_pose](#), 177
  - [updatePose](#), 177
- [OFIQ\\_LIB::SegmentationExtractorInterface](#), 181
  - [~SegmentationExtractorInterface](#), 182
  - [GetLastSessionId](#), 182
  - [GetMask](#), 182
  - [m\\_lastSessionId](#), 183
  - [m\\_masks](#), 183
  - [UpdateMask](#), 182
- [OFIQ\\_LIB::Session](#), 183
  - [assessment](#), 186
  - [GenerateId](#), 186
  - [getAlignedFace](#), 186
  - [getAlignedFaceLandmarkedRegion](#), 186
  - [getAlignedFaceLandmarks](#), 186
  - [getAlignedFaceTransformationMatrix](#), 187
  - [getDetectedFaces](#), 187
  - [getFaceOcclusionSegmentationImage](#), 187
  - [getFaceParsingImage](#), 187
  - [getLandmarks](#), 187
  - [getPose](#), 188
  - [Id](#), 188
  - [image](#), 188
  - [m\\_alignedFace](#), 191
  - [m\\_alignedFacelandmarkedRegion](#), 191
  - [m\\_alignedFaceLandmarks](#), 191
  - [m\\_alignedFaceTransformationMatrix](#), 191
  - [m\\_assessment](#), 191
  - [m\\_detectedFaces](#), 191
  - [m\\_faceOcclusionSegmentationImage](#), 191
  - [m\\_faceParsingImage](#), 192
  - [m\\_id](#), 192
  - [m\\_image](#), 192
  - [m\\_landmarks](#), 192
  - [m\\_pose](#), 192
  - [Session](#), 185
  - [setAlignedFace](#), 188
  - [setAlignedFaceLandmarkedRegion](#), 189
  - [setAlignedFaceLandmarks](#), 189
  - [setAlignedFaceTransformationMatrix](#), 189
  - [setDetectedFaces](#), 189
  - [setFaceOcclusionSegmentationImage](#), 190
  - [setFaceParsingImage](#), 190
  - [setLandmarks](#), 190
  - [setPose](#), 190
- [ofiq\\_lib\\_impl.h](#), 211, 212
- [ofiq\\_structs.h](#), 213, 215
- [OFIQError](#)
  - [OFIQ\\_LIB::OFIQError](#), 162
- [OFIQError.h](#), 285, 286
- [OFIQImpl](#)
  - [OFIQ\\_LIB::OFIQImpl](#), 164
- [ONNXRTSegmentation.h](#), 276, 277
- [ONNXRuntimeSegmentation](#), 168
  - [~ONNXRuntimeSegmentation](#), 169
  - [getNumberOfOutputNodes](#), 169
  - [init\\_session](#), 169
  - [initialize](#), 170
  - [m\\_inputShape](#), 170
  - [m\\_memoryInfo](#), 170



- m\_ortenv, 170
  - m\_ortSession, 170
  - ONNXRuntimeSegmentation, 169
  - run, 170
- Open Source Face Image Quality (OFIQ) Library, 1
- opencv\_ssd\_face\_detector.h, 220, 221
- OPENCVSSD
  - OFIQ, 43
- operator<<
  - OFIQ, 45
- OverExposurePrevention
  - OFIQ, 43
  - OFIQ\_LIB::modules::measures::OverExposurePrevention, 172
- OverExposurePrevention.h, 262, 263
- pairsInnerLip
  - OFIQ\_LIB::modules::landmarks::adnet, 61
- pairsLeftEye
  - OFIQ\_LIB::modules::landmarks::adnet, 61
- pairsMouthCenter
  - OFIQ\_LIB::modules::landmarks::adnet, 61
- pairsRightEye
  - OFIQ\_LIB::modules::landmarks::adnet, 62
- parameters
  - OFIQ\_LIB::Configuration, 82
- PartExtractor.h, 231, 232
- performPreprocessing
  - OFIQ\_LIB::OFIQImpl, 166
- Point2f, 174
  - x, 175
  - y, 175
- poseEstimator
  - OFIQ\_LIB::NeuronalNetworkContainer, 157
- poseEstimators.h, 271, 272
- qAssessments
  - OFIQ::FacelImageQualityAssessment, 99
- QualityAssessmentError
  - OFIQ, 45
- QualityAssessments
  - OFIQ, 42
- QualityMeasure
  - OFIQ, 43
- QualityMeasureResult
  - OFIQ::QualityMeasureResult, 178
- QualityMeasureReturnCode
  - OFIQ, 44
- r\_brow
  - OFIQ\_LIB::modules::segmentations, 65
- r\_ear
  - OFIQ\_LIB::modules::segmentations, 66
- r\_eye
  - OFIQ\_LIB::modules::segmentations, 65
- rawScore
  - OFIQ::QualityMeasureResult, 179
- readImage
  - OFIQ\_LIB, 53
- readImageFromByteArray
  - OFIQ\_LIB, 55
- ReduceImageToRegionOfInterest
  - OFIQ\_LIB::modules::measures::NaturalColour, 155
- Reset
  - OFIQ\_LIB::modules::measures::SigmoidParameters, 198
- ReturnCode
  - OFIQ, 44
- ReturnStatus
  - OFIQ::ReturnStatus, 180
- RIGHT\_EYE
  - OFIQ\_LIB::modules::landmarks, 58
- RIGHT\_EYE\_CORNERS
  - OFIQ\_LIB::modules::landmarks, 58
- rightEye
  - OFIQ\_LIB::modules::landmarks::adnet, 62
- rightEyeCorners
  - OFIQ\_LIB::modules::landmarks::adnet, 62
- RightwardCropOfTheFacelImage
  - OFIQ, 44
- round
  - OFIQ\_LIB::modules::measures::SigmoidParameters, 198
- run
  - ONNXRuntimeSegmentation, 170
- s
  - OFIQ\_LIB::modules::measures::SigmoidParameters, 199
- scalar
  - OFIQ::QualityMeasureResult, 179
- ScalarConversion
  - OFIQ\_LIB::modules::measures::Measure, 145
- scalarQuality
  - OFIQ::Interface, 134
  - OFIQ\_LIB::OFIQImpl, 166
- segmentationExtractor
  - OFIQ\_LIB::NeuronalNetworkContainer, 157
- segmentations.h, 277, 279
- SegmentClassLabels
  - OFIQ\_LIB::modules::segmentations, 65
- Session
  - OFIQ\_LIB::Session, 185
- Session.h, 286, 287
- setAlignedFace
  - OFIQ\_LIB::Session, 188
- setAlignedFaceLandmarkedRegion
  - OFIQ\_LIB::Session, 189
- setAlignedFaceLandmarks
  - OFIQ\_LIB::Session, 189
- setAlignedFaceTransformationMatrix
  - OFIQ\_LIB::Session, 189
- SetDataDir
  - OFIQ\_LIB::Configuration, 82
- setDetectedFaces
  - OFIQ\_LIB::Session, 189
- setFaceOcclusionSegmentationImage



- OFIQ\_LIB::Session, 190
- setFaceParsingImage
  - OFIQ\_LIB::Session, 190
- SetImage
  - OFIQ\_LIB::modules::segmentations::FaceParsing, 115
- setInverse
  - OFIQ\_LIB::modules::measures::SigmoidParameters, 198
- setLandmarks
  - OFIQ\_LIB::Session, 190
- setPose
  - OFIQ\_LIB::Session, 190
- SetQualityMeasure
  - OFIQ\_LIB::modules::measures::Measure, 145
- Sharpness
  - OFIQ, 44
  - OFIQ\_LIB::modules::measures::Sharpness, 194
- Sharpness.h, 263, 264
- Sigmoid
  - OFIQ\_LIB::modules::measures::Measure, 146
- SigmoidParameters
  - OFIQ\_LIB::modules::measures::SigmoidParameters, utils.h, 288, 290 198
- SingleFacePresent
  - OFIQ, 44
  - OFIQ\_LIB::modules::measures::SingleFacePresent, 201
- SingleFacePresent.h, 265, 266
- size
  - OFIQ::Image, 129
- skin
  - OFIQ\_LIB::modules::segmentations, 65
- SSDFaceDetector
  - OFIQ\_LIB::modules::detectors::SSDFaceDetector, 202
- Success
  - OFIQ, 44, 45
- tmetric
  - OFIQ\_LIB, 55
- type
  - OFIQ::FaceLandmarks, 101
- u\_lip
  - OFIQ\_LIB::modules::segmentations, 66
- UnderExposurePrevention
  - OFIQ, 43
  - OFIQ\_LIB::modules::measures::UnderExposurePrevention, 205
- UnderExposurePrevention.h, 266, 267
- UnifiedQualityScore
  - OFIQ, 43
  - OFIQ\_LIB::modules::measures::UnifiedQualityScore, ytop 207
- UnifiedQualityScore.h, 267, 268
- UnknownConfigParamError
  - OFIQ, 45
- UnknownError
  - OFIQ, 45
- UpdateFaces
  - OFIQ\_LIB::FaceDetectorInterface, 97
  - OFIQ\_LIB::modules::detectors::SSDFaceDetector, 203
- updateLandmarks
  - OFIQ\_LIB::FaceLandmarkExtractorInterface, 100
  - OFIQ\_LIB::modules::landmarks::ADNetFaceLandmarkExtractor, 68
- UpdateMask
  - OFIQ\_LIB::modules::segmentations::FaceOcclusionSegmentation, 110
  - OFIQ\_LIB::modules::segmentations::FaceParsing, 116
  - OFIQ\_LIB::SegmentationExtractorInterface, 182
- updatePose
  - OFIQ\_LIB::modules::poseEstimators::HeadPose3DDFAV2, 122
  - OFIQ\_LIB::PoseEstimatorInterface, 177
- Upper
  - OFIQ\_LIB::modules::landmarks::LandmarkPair, 136
- vectorQuality
  - OFIQ::Interface, 134
  - OFIQ\_LIB::OFIQImpl, 166
- w
  - OFIQ\_LIB::modules::measures::SigmoidParameters, 199
- what
  - OFIQ\_LIB::OFIQError, 162
- whatCode
  - OFIQ\_LIB::OFIQError, 162
- width
  - OFIQ::BoundingBox, 74
  - OFIQ::Image, 130
- x
  - OFIQ::LandmarkPoint, 137
  - OFIQ\_LIB::Point2i, 175
  - Point2f, 175
- x0
  - OFIQ\_LIB::modules::measures::SigmoidParameters, 199
- xleft
  - OFIQ::BoundingBox, 74
- y
  - OFIQ::LandmarkPoint, 137
  - OFIQ\_LIB::Point2i, 175
  - Point2f, 175
- ytop
  - OFIQ::BoundingBox, 74