My Project

Generated by Doxygen 1.9.1

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Class Documentation

3.1 BufferWriter Class Reference

Public Member Functions

- void **write** (std::vector< Eigen::VectorXf > &buffer, std::vector< std::chrono::system_clock::time_point > &peakTimesBuffer, const std::string &outputFile)

Public Attributes

- std::chrono::milliseconds **_flushInterval**
- size_t **_bufferSizeThreshold**
- std::chrono::time_point< std::chrono::steady_clock > **_lastFlushTime**

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

- C/src/buffer_writter.h

3.2 DetectionResult Struct Reference

Public Attributes

- int **minPeakIndex** = -1
- int **maxPeakIndex** = -1
- std::chrono::system_clock::time_point **peakTimes**
- float **peakAmplitude**

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

- C/src/process_data.h

3.3 ExperimentConfig Class Reference

Public Attributes

- const std::function< void(std::span< float >, Eigen::MatrixXf &, unsigned int)> **ProcessFncPtr** = ProcessSegmentInterleaved
- std::unique_ptr< ONNXModel > **onnxModel**

Static Public Attributes

- static constexpr int **HEAD_SIZE** = 12
- static constexpr int **NUM_CHAN** = 4
- static constexpr int **SAMPS_PER_CHANNEL** = 124
- static constexpr int **BYTES_PER_SAMP** = 2
- static constexpr int **MICRO_INCR** = 1240
- static constexpr int **SAMPLE_RATE** = 1e5
- static constexpr int **DATA_SIZE** = SAMPS_PER_CHANNEL * NUM_CHAN * BYTES_PER_SAMP
- static constexpr int **PACKET_SIZE** = DATA_SIZE + HEAD_SIZE
- static constexpr int **REQUIRED_BYTES** = DATA_SIZE + HEAD_SIZE
- static constexpr int **DATA_BYTES_PER_CHANNEL** = SAMPS_PER_CHANNEL * BYTES_PER_SAMP
- static constexpr float **TIME_WINDOW** = 0.01
- static constexpr int **NUM_PACKS_DETECT** = static_cast<int>(TIME_WINDOW * 100000 / SAMPS_PER_CHANNEL)
- static constexpr int **DATA_SEGMENT_LENGTH** = NUM_PACKS_DETECT * SAMPS_PER_CHANNEL * NUM_CHAN
- static constexpr float **speedOfSound** = 1482.965459
- static constexpr int **interp** = 1
- static constexpr const char * **filterWeights** = "filters/highpass_taps@101_cutoff@20k_window@hamming_fs@100k.txt"
- static constexpr const char * **receiverPositions** = "../Data/SOCAL_H_72_HS_harp4chPar_recPos.txt"
- static constexpr const char * **onnxModelPath** = "../TestOnnx/model_quantized_static.onnx"
- static constexpr const char * **onnxModelScaling** = "../TestOnnx/scaler_params.json"

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

- C/src/custom_types.h

3.4 ExperimentRuntime Class Reference

Public Attributes

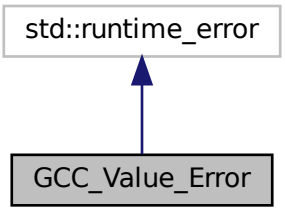
- std::string **detectionOutputFile** = ""
- std::chrono::seconds **programRuntime**
- TimePoint **programStartTime**
- float **energyDetThresh** = 100.0f
- fftwf_plan **forwardFFT** = nullptr
- fftwf_plan **inverseFFT** = nullptr

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

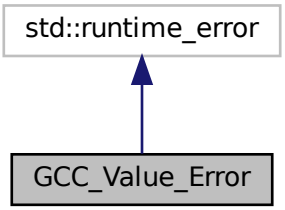
- C/src/custom_types.h

3.5 GCC_Value_Error Class Reference

Inheritance diagram for GCC_Value_Error:



Collaboration diagram for GCC_Value_Error:



Public Member Functions

- **GCC_Value_Error** (const std::string &message)

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

- C/src/custom_types.h

3.6 ONNXModel Class Reference

Public Member Functions

- **ONNXModel** (const std::string &model_path, const std::string &scaler_params_path)
- std::vector< int64_t > **get_input_node_info** ()
- void **load_scaler_params** (const std::string &file_path)
- std::vector< float > **run_inference** (const std::vector< float > &input_tensor_values)
- void **normalize_data** (std::vector< float > &data) const

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

- C/src/onnx_model.h

3.7 Session Class Reference

Public Member Functions

- int **pushDataToBuffer** (const std::vector< uint8_t > &data)
- std::vector< uint8_t > **popDataFromBuffer** ()

Public Attributes

- std::atomic< bool > **errorOccurred** = false
- std::queue< std::vector< uint8_t > > **dataBuffer**
- std::vector< std::vector< uint8_t > > **dataBytesSaved**
- std::vector< float > **dataSegment**
- std::vector< std::chrono::system_clock::time_point > **dataTimes**
- std::mutex **dataBufferLock**
- int **detectionCounter** = 0
- std::vector< Eigen::VectorXf > **Buffer**
- std::vector< std::chrono::system_clock::time_point > **peakTimesBuffer**

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

- C/src/session.h

3.8 SocketManager Class Reference

Public Member Functions

- void RestartListener ()

Public Attributes

- int **datagramSocket** = socket(AF_INET, SOCK_DGRAM, 0)
- int **UDP_PORT**
- std::string **UDP_IP**

3.8.1 Member Function Documentation

3.8.1.1 RestartListener()

```
void SocketManager::RestartListener ( )  [inline]
```

(Re)starts the udp listener. It closes the existing socket connection and creates a new one. Additionally, it clears the buffer (dataBuffer) and the segment to be processed (dataSegment) as well as the vector containing the timestamps (dataTimes).

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

- C/src/socket_manager.h

Index