Index

Module **audioprocessing**

▸ EXPAND SOURCE CODE

Classes

```
class AudioProcessor (parameters: dict)
```

Abstract class to handle all Audio Processing methods and functions. Can be inherited to implement custom processing methods.

Constructor for the AudioProcessor class.

Args:

`parameters` : A dictionary containing audio parameters.

Class Attributes:

`_sample_rate` : Sample rate at which to read and write audio.

`_chunk_size` : Size of the chunk of audio to read.

`_np_format` : Numpy format used to process audio.

`_nfft` : Size of the FFT used during processing.

`_hop_length` : Hop Length of the FFT used during processing.

`_window_size` : Window Size of the FFT used during processing.

`_window_type` : Window Type of the FFT used during processing.

`_p_threshold` : Threshold under which the frequency component of the audio piece processed during polyphonic pitch extraction is discarded.

`_normType` : Normalization type used during processing.

▸ EXPAND SOURCE CODE

Ancestors

abc.ABC

Subclasses

DefaultAudioProcessor

Methods

```
def process(self, frame, inst: modules.utilities.Instruments)
```

Abstract method to process a given audio frame.

Args:

`frame` : Chunk of audio to process.

`inst` : Instrument assigned to the track (used to get the frequency range for audio processing).

▸ EXPAND SOURCE CODE

```
class DefaultAudioProcessor (parameters: dict)
```

Implements the default chain used to process low-level features.

Constructor for the DefaultAudioProcessor class.

Args:

`parameters` : A dictionary containing audio parameters.

▸ EXPAND SOURCE CODE

Ancestors

AudioProcessor, abc.ABC

Methods

```
def process(self, frame, inst: modules.utilities.Instruments)
```

Processes the given audio frame according to a defined chain.

Args:

`frame` : Audio frame to process.

`inst` : Instrument of the track from which to get the frequency range.

Returns:

Low-level Features as floats ordered inside an array.

▸ EXPAND SOURCE CODE

```
class HFAudioInputHandler (parameters: dict, channel: int,
                           instrument: modules.utilities.Instruments)
```

Handles the High-level feature processing of either live or recorded audio.

Constructor of the HFAudioInputHandler class.

Args:

`parameters` : Audio parameters used to process the audio.

`channel` : Unused (high-level features are computed over the sum of all tracks).

`instrument` : Unused (high-level features are computed over the sum of all tracks).

Class Attributes:

`__arousal_values` : Array containing previous arousal values used to compute a moving average.

`__valence_values` : Array containing previous valence values used to compute a moving average.

`__nn_model` : Neural network model used to extract the mood from the piece of audio.

▸ EXPAND SOURCE CODE

Ancestors

InputHandler, abc.ABC

Methods

```
def process(self, data)
```

Processes an audio frame for High-level features.

Args:

`data` : Piece of audio to process.

▸ EXPAND SOURCE CODE

Inherited members

`InputHandler`: `get_instrument`, `get_priority`, `handle_settings`, `set_instrument`, `set_priority`

```
class InputHandler (parameters: dict, channel: int,
                    instrument: modules.utilities.Instruments)
```

The InputHandler abstract class declares a set of methods for processing data and extract features. Can be inherited to implement custom methods to process data and extract features.

Constructor for the InputHandler class.

Args:

`parameters` : Audio parameters used to process the audio.

`channel` : Channel index of the track to process assigned to this instance.

`instrument` : Instrument of the track to process assigned to this instance.

Class Attributes:

`_connection_handler` : Object used to send features to external applications.

`_signal_threshold` : Threshold under which the signal is considered to be null.

`__instrument` : Instrument of the source of audio.

`_lock` : Mutex lock used for synchronization purposes.

`channel` : Number assigned to the track that this object is currently processing.

`__priority` : Number to indicate the priority value of an instance with respect to the other instances.

▸ EXPAND SOURCE CODE

Ancestors

abc.ABC

Subclasses

HFAudioInputHandler, LFAudioInputHandler

Methods

```
def get_instrument(self) -> modules.utilities.Instruments
```

Synchronized getter for the `__instrument` attribute.

Returns:

The instrument assigned to this handler.

▸ EXPAND SOURCE CODE

```
def get_priority(self) -> int
```

Synchronized getter for the `__priority` attribute.

Returns:

The priority assigned to this handler.

▸ EXPAND SOURCE CODE

```
def handle_settings(self, settings)
```

Handles incoming settings.

Args:

`settings` : settings to handle.

▸ EXPAND SOURCE CODE

```
def process(self, data)
```

Abstract method to implement the processing and feature extraction chain.

Args:

`data` : Data to process.

▸ EXPAND SOURCE CODE

```
def set_instrument(self, instrument: modules.utilities.Instruments)
```

Synchronized setter for the `__instrument` attribute.

Args:

`instrument` : New instrument to assign.

▸ EXPAND SOURCE CODE

```
def set_priority(self, priority: int)
```

Synchronized setter for the `__priority` attribute.

Args:

`priority` : New priority to assign.

▸ EXPAND SOURCE CODE

```
class LFAudioInputHandler (parameters: dict, channel: int,
                           instrument: modules.utilities.Instruments)
```

Handles the Low-level feature processing of a channel of either live or recorded audio.

Constructor for the LFAudioInputHandler class.

Args:

`parameters` : Audio parameters used to process the audio.

`channel` : Channel index of the track to process assigned to this instance.

`instrument` : Instrument of the track to process assigned to this instance.

Class Attributes:

`_audio_processor` : Processor object used to process audio.

▸ EXPAND SOURCE CODE

Ancestors

InputHandler, abc.ABC

Methods

```
def process(self, data)
```

Processes an audio frame or returns if the frame has virtually no signal.

Args:

`data` : Data to process.

▸ EXPAND SOURCE CODE

Inherited members

`InputHandler`: `get_instrument`, `get_priority`, `handle_settings`, `set_instrument`, `set_priority`