Python python_speech_features.mfcc() Examples

The following are code examples for showing how to use *python_speech_features.mfcc()*. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

Example 1

Project: *AudioEmotionDetection* Author: *DefinitelyNotTim* File: *emotionProcessor.py* MIT License 6 votes

```
def __exit__(self, exception, value, traceback):
        self.close()


##   mfccProc: extracts the MFCCs from given audio
##   Written by Timmothy Lane
##   Creates 2d arrays for storage of the fbank feature, mfcc features
##   and the delta of MFCC features
##   NOTE: code used to create 2 dimensional arrays for both the delta
##      of MFCCs and log of the filterbank features are included, but commented
##      out. These statements were included for use by future researchers who
##      may want to experiment with the different metrics to improce accuracy.
##  Inputs: self
##  Output: an array containing the Mel-Frequency Cepstrum Coefficients
##  Related Software Requirements: FR.6
##  Author: Timmothy Lane
```

Example 2

Project: *AudioEmotionDetection* Author: *DefinitelyNotTim* File: *emotionProcessor.py* MIT License 6 votes

```
def mfccProc(self):
        (rate,sig) = audioBasicIO.readAudioFile(self.fname)
        #Create 2d array for MFCC features
        mfcc_feat = mfcc(sig,samplerate = 44100, nfft = 1103)
        #Create 2d array for the delta of MFCC features
        #d_mfcc_feat = delta(mfcc_feat, 2)
        #Create 2d array for the log of fbank features
        #fbank_feat = logfbank(sig,rate)
                #Return the Mel-Frequency Cepstrum Coefficients
        return(mfcc_feat)


## Description: This function extracts the pitch of the audio file and puts it into a list of short term measurements from the file
## Inputs:self
## Outputs: array of the pitches for the short term chunks
##  Related Software Requirements: FR.3
##Author: Alex Shannon
```

Example 3

Project: *Speech-commands-recognition* Author: *lucko515* File: *audio_utils.py* MIT License 6 votes

```
def compute_mfcc_features(audio_sample, sample_rate, numcep=13):
    '''
    Computes MFCC, delta MFCC and delta-delta MFCC features of the input audio sample.

    :params:
        audio_sample, numpy array, raw audio signal (example: in wav format)
        sample_rate, Integer
        numcap - Integer, the number of cepstrum features to return

    :returns:
        mfcc_features, Numpy array,
        delta1_mfcc, numpy array,
        delta2_mfcc, numpy array,
    '''

    mfcc_features = mfcc(audio_sample, sample_rate, numcep=numcep)

    delta1_mfcc = librosa.feature.delta(mfcc_features, order=1)

    delta2_mfcc = librosa.feature.delta(mfcc_features, order=2)

    return mfcc_features, delta1_mfcc, delta2_mfcc
```

Example 4

Project: *mist-rnns* Author: *rdipietro* File: *timitphonemerec.py* Apache License 2.0 6 votes

```
def _mfcc_and_labels(audio, labels):
  """ Convert to MFCC features and corresponding (interpolated) labels.

  Returns:
    A tuple, `(mfcc_features, mfcc_labels)`. A 1-D float array and a 1-D int
      array, both with the same shape.
  """
  mfcc_sample_rate = 100.0
  winfunc = lambda x: np.hamming(x)
  mfcc_features = python_speech_features.mfcc(audio, samplerate=timit.SAMPLE_RATE, winlen=0.025,
                                              winstep=1.0/mfcc_sample_rate, lowfreq=85.0,
                                              highfreq=timit.SAMPLE_RATE/2, winfunc=winfunc)
  t_audio = np.linspace(0.0, audio.shape[0] * 1.0 / timit.SAMPLE_RATE, audio.size, endpoint=False)
  t_mfcc = np.linspace(0.0, mfcc_features.shape[0] * 1.0 / mfcc_sample_rate, mfcc_features.shape[0], endpoint=False)
  interp_func = scipy.interpolate.interp1d(t_audio, labels, kind='nearest')
  mfcc_labels = interp_func(t_mfcc)
  return mfcc_features, mfcc_labels
```

Example 5

Project: *Voiceprint-Recognition*   Author: *SunYanCN*   File: *mfcc.py*   Apache License 2.0   6 votes

```
def generate_mfcc(sig, rate, sig_len, noise=None, noise_weight=0.1, winlen=0.03125, winstep=0.03125/2, numcep=13, nfilt=26, nfft=512
    if(len(sig) != sig_len):
        if(len(sig)< sig_len):
            sig = np.pad(sig, (0, sig_len - len(sig)), 'constant')
        if(len(sig) >sig_len):
            sig = sig[0:sig_len]
    # i dont know, 'tensorflow' normalization
    sig = sig.astype('float') / 32768

    if(noise is not None):
        noise = noise[random.randint(0, len(noise)-1)] # pick a noise
        start = random.randint(0, len(noise)-sig_len) # pick a sequence
        noise = noise[start:start+sig_len]
        noise = noise.astype('float')/32768
        sig = sig * (1-noise_weight) + noise * noise_weight
        #wav.write('noise_test.wav', rate, sig)
    mfcc_feat = mfcc(sig, rate, winlen=winlen, winstep=winstep, numcep=numcep, nfilt=nfilt, nfft=nfft, lowfreq=lowfreq,
                     highfreq=highfreq, winfunc=winfunc, ceplifter=ceplifter, preemph=preemph)
    mfcc_feat = mfcc_feat.astype('float32')
    return mfcc_feat
```

Example 6

Project: *Voiceprint-Recognition*   Author: *SunYanCN*   File: *mfcc.py*   Apache License 2.0   6 votes

```
def generate_mfcc(sig, rate, sig_len, noise=None, noise_weight=0.1, winlen=0.03125, winstep=0.03125/2, numcep=13, nfilt=26, nfft=512
    if(len(sig) != sig_len):
        if(len(sig)< sig_len):
            sig = np.pad(sig, (0, sig_len - len(sig)), 'constant')
        if(len(sig) >sig_len):
            sig = sig[0:sig_len]

    # i dont know, 'tensorflow' normalization
    sig = sig.astype('float') / 32768

    if(noise is not None):
        noise = noise[random.randint(0, len(noise)-1)] # pick a noise
        start = random.randint(0, len(noise)-sig_len) # pick a sequence
        noise = noise[start:start+sig_len]
        noise = noise.astype('float')/32768
        sig = sig * (1-noise_weight) + noise * noise_weight
        #wav.write('noise_test.wav', rate, sig)
    mfcc_feat = mfcc(sig, rate, winlen=winlen, winstep=winstep, numcep=numcep, nfilt=nfilt, nfft=nfft, lowfreq=lowfreq,
                     highfreq=highfreq, winfunc=winfunc, ceplifter=ceplifter, preemph=preemph)
    mfcc_feat = mfcc_feat.astype('float32')
    return mfcc_feat
```

Example 7

Project: *Audio-Genre-Classification*   Author: *dhruvesh13*   File: *tester.py*   MIT License   6 votes

```
def create_ceps_test(fn):
    """
        Creates the MFCC features from the test files,
        saves them to disk, and returns the saved file name.
    """
    sample_rate, X = scipy.io.wavfile.read(fn)
    # X[X==0]=1
    # np.nan_to_num(X)
    ceps= mfcc(X)
    bad_indices = np.where(np.isnan(ceps))
    b=np.where(np.isinf(ceps))
    ceps[bad_indices]=0
    ceps[b]=0
    base_fn, ext = os.path.splitext(fn)
    data_fn = base_fn + ".ceps"
    np.save(data_fn, ceps)
    print "Written ", data_fn
    return data_fn
```

Example 8

6 votes

Project: *Audio-Genre-Classification* Author: *dhruvesh13* File: *mfcc-features.py* MIT License 👍 👎

```
def create_ceps(wavfile):
        sampling_rate, song_array = scipy.io.wavfile.read(wavfile)
        print(sampling_rate)
        """Get MFCC
        ceps  : ndarray of MFCC
        mspec : ndarray of log-spectrum in the mel-domain
        spec  : spectrum magnitude
        """
        ceps=mfcc(song_array)
        #ceps, mspec, spec= mfcc(song_array)
        print(ceps.shape)
        #this is done in order to replace NaN and infinite value in array
        bad_indices = np.where(np.isnan(ceps))
        b=np.where(np.isinf(ceps))
        ceps[bad_indices]=0
        ceps[b]=0
        write_ceps(ceps, wavfile)

# Saves mfcc data
```

Example 9

Project: *GlosBio* Author: *MikolajBalcerek* File: *mfcc_plot.py* MIT License 6 votes 👍 👎

```
def _plot_mfcc_color_boxes(audio_path: str) -> plt.Figure:
    """
    Plots a MFCC figure from an audio file (wav) under audio_path

    :param audio_path: str full path to audio file
    :return: plt.Figure containing the MFCC colored boxes plot
    """
    (rate, sig) = wav.read(audio_path)
    mfcc_features_lines = mfcc(sig, rate, nfft=1250)
    figure, axis = plt.subplots()
    mfcc_data = np.swapaxes(mfcc_features_lines, 0, 1)
    axis.set_title("MFCC")
    format_figure = axis.imshow(mfcc_data, interpolation='nearest', cmap=cm.coolwarm,
                                origin='lower', aspect='auto')

    return figure
```

Example 10

Project: *GlosBio* Author: *MikolajBalcerek* File: *mfcc_plot.py* MIT License 6 votes 👍 👎

```
def _plot_mfcc_color_boxes_from_bytes(audio_bytes: bytes) -> plt.Figure:
    """
    Plots a MFCC figure from an audio file (wav) under audio_path

    :param audio_path: str full path to audio file
    :return: plt.Figure containing the MFCC colored boxes plot
    """
    plt.clf()
    (rate, sig) = wav.read(audio_bytes)
    mfcc_features_lines = mfcc(sig, rate, nfft=1250)
    figure, axis = plt.subplots()
    mfcc_data = np.swapaxes(mfcc_features_lines, 0, 1)
    axis.set_title("MFCC")
    format_figure = axis.imshow(mfcc_data, interpolation='nearest', cmap=cm.coolwarm,
                                origin='lower', aspect='auto')

    return figure
```

Example 11

Project: *MTGAN* Author: *zengchang94622* File: *dataset.py* MIT License 6 votes 👍 👎

```
def make_feature(y, sr):
    if FEATURE == 'fft':
        S = librosa.stft(y, n_fft = N_FFT, hop_length = HOP_LEN, window = hamming)
        feature, _ = librosa.magphase(S)
        feature = np.log1p(feature)
        feature = feature.transpose()
    else:
        if FEATURE == 'fbank':
            feature = logfbank(y, sr, nfilt = FEATURE_LEN, winlen = WIN_LEN, winstep = WIN_STEP)
            assert feature.shape[-1] == FEATURE_LEN, '{}'.format(feature.shape[-1])
        else:
            feature =  mfcc(y, sr, nfilt = FEATURE_LEN, winlen = WIN_LEN, winstep = WIN_STEP)
            feature_d1 = delta(feature, N = 1)
            feature_d2 = delta(feature, N = 2)
            feature = np.hstack([feature, feature_d1, feature_d2])
    return normalize(feature).astype(np.float32)
```

Example 12

6 votes

Project: *tensorflow-ctc-speech-recognition* Author: *philipperemy* File: *utils.py* Apache License 2.0

```
def convert_inputs_to_ctc_format(audio, fs, target_text, num_features):
    # print(target_text)
    inputs = mfcc(audio, samplerate=fs, numcep=num_features)
    # Transform in 3D array
    train_inputs = np.asarray(inputs[np.newaxis, :])
    train_inputs = (train_inputs - np.mean(train_inputs)) / np.std(train_inputs)
    train_seq_len = [train_inputs.shape[1]]

    # Get only the words between [a-z] and replace period for none
    original = ' '.join(target_text.strip().lower().split(' ')).replace('.', '').replace('?', '').replace(',',
                                                                                                          '').replace(
        "'", '').replace('!', '').replace('-', '')
    # print(original)
    targets = original.replace(' ', '  ')
    targets = targets.split(' ')

    # Adding blank label
    targets = np.hstack([SPACE_TOKEN if x == '' else list(x) for x in targets])

    # Transform char into index
    targets = np.asarray([SPACE_INDEX if x == SPACE_TOKEN else ord(x) - FIRST_INDEX
                          for x in targets])

    return train_inputs, targets, train_seq_len, original
```

Example 13

Project: *speech-driven-hand-gesture-generation-demo* Author: *Svito-zar* File: *tools.py* Apache License 2.0 6 votes

```
def calculate_mfcc(audio_filename):
    """
    Calculate MFCC features for the audio in a given file
    Args:
        audio_filename: file name of the audio

    Returns:
        feature_vectors: MFCC feature vector for the given audio file
    """
    fs, audio = wav.read(audio_filename)

    # Make stereo audio being mono
    if len(audio.shape) == 2:
        audio = (audio[:, 0] + audio[:, 1]) / 2

    # Calculate MFCC feature with the window frame it was designed for
    feature_vectors = mfcc(audio, winlen=0.06666666666, winstep=0.0166666666, samplerate=fs, numcep=MFCC_INPUTS)

    # Subsample if nesessary
    if SUBSAMPL_RATE > 3:
        feature_vectors = feature_vectors[0::3]

    return feature_vectors
```

Example 14

Project: *Speaker_Verification_Tencent* Author: *zengchang94622* File: *dataset.py* MIT License 6 votes

```
def make_feature(y, sr):
    if FEATURE == 'fft':
        S = librosa.stft(y, n_fft = N_FFT, hop_length = HOP_LEN, window = hamming)
        feature, _ = librosa.magphase(S)
        feature = np.log1p(feature)
        feature = feature.transpose()
    else:
        if FEATURE == 'fbank':
            feature = logfbank(y, sr, nfilt = FEATURE_LEN, winlen = WIN_LEN, winstep = WIN_STEP)
            assert feature.shape[-1] == FEATURE_LEN, '{}'.format(feature.shape[-1])
        else:
            feature = mfcc(y, sr, nfilt = FEATURE_LEN, winlen = WIN_LEN, winstep = WIN_STEP)
            feature_d1 = delta(feature, N = 1)
            feature_d2 = delta(feature, N = 2)
            feature = np.hstack([feature, feature_d1, feature_d2])
    return normalize(feature).astype(np.float32)
```

Example 15

Project: *ip9* Author: *tiefenauer* File: *corpus2csv.py* MIT License 6 votes

```
def create_subset(h5_file, name, df):
    h5_file.create_dataset(f'{name}/features', shape=(0,), maxshape=(None,), dtype=h5py.special_dtype(vlen=np.float32))
    h5_file.create_dataset(f'{name}/labels', shape=(0,), maxshape=(None,), dtype=h5py.special_dtype(vlen=str))
    h5_file.create_dataset(f'{name}/durations', shape=(0,), maxshape=(None,))
    progress = tqdm(zip(df['wav_filename'], df['wav_filesize'], df['transcript']), total=len(df.index))
    for wav_file_path, wav_file_size, transcript in progress:
        progress.set_description(f'{name}: {wav_file_path}')
        inputs = h5_file[name]['features']
        labels = h5_file[name]['labels']
        durations = h5_file[name]['durations']

        rate, audio = wavfile.read(wav_file_path)
        inp = mfcc(audio, samplerate=rate, numcep=26)  # (num_timesteps x num_features)
        inputs.resize(inputs.shape[0] + 1, axis=0)
        inputs[inputs.shape[0] - 1] = inp.flatten().astype(np.float32)

        labels.resize(labels.shape[0] + 1, axis=0)
        labels[labels.shape[0] - 1] = transcript

        durations.resize(durations.shape[0] + 1, axis=0)
        durations[durations.shape[0] - 1] = wav_file_size
```

Example 16

Project: *Speech_driven_gesture_generation_with_autoencoder* Author: *GestureGeneration*
File: *tools.py* Apache License 2.0

6 votes 👍 👎

```
def calculate_mfcc(audio_filename):
    """
    Calculate MFCC features for the audio in a given file
    Args:
        audio_filename: file name of the audio

    Returns:
        feature_vectors: MFCC feature vector for the given audio file
    """
    fs, audio = wav.read(audio_filename)

    # Make stereo audio being mono
    if len(audio.shape) == 2:
        audio = (audio[:, 0] + audio[:, 1]) / 2

    # Calculate MFCC feature with the window frame it was designed for
    input_vectors = mfcc(audio, winlen=0.02, winstep=0.01, samplerate=fs, numcep=MFCC_INPUTS)

    input_vectors = [average(input_vectors[:, i], 5) for i in range(MFCC_INPUTS)]

    feature_vectors = np.transpose(input_vectors)

    return feature_vectors
```

Example 17

Project: *Speaker-Identification-Python* Author: *Atul-Anand-Jha* File: *featureextraction.py* GNU Lesser General Public License v3.0

5 votes 👍 👎

```
def extract_features(audio,rate):
    """extract 20 dim mfcc features from an audio, performs CMS and combines
    delta to make it 40 dim feature vector"""

    mfcc_feature = mfcc.mfcc(audio,rate, 0.025, 0.01,20,nfft = 1200, appendEnergy = True)
    mfcc_feature = preprocessing.scale(mfcc_feature)
    delta = calculate_delta(mfcc_feature)
    combined = np.hstack((mfcc_feature,delta))
    return combined
```

Example 18

Project: *AudioEmotionDetection* Author: *DefinitelyNotTim* File: *emotionProcessor-threaded.py* MIT License

5 votes 👍 👎

```
def __exit__(self, exception, value, traceback):
        self.close()

#mfccProc: extracts the MFCCs from given audio
# Written by Timmothy Lane
# Creates 2d arrays for storage of the fbank feature, mfcc features
#   and the delta of MFCC features
# Written By: Timmothy Lane
```

Example 19

Project: *AudioEmotionDetection* Author: *DefinitelyNotTim* File: *emotionProcessor-threaded.py* MIT License

5 votes 👍 👎

```
def mfccProc(self):
        (rate,sig) = audioBasicIO.readAudioFile(self.fname)
        #Create 2d array for MFCC features
        mfcc_feat = mfcc(sig,samplerate = 44100, nfft = 1103)
        #Create 2d array for the delta of MFCC features
        d_mfcc_feat = delta(mfcc_feat, 2)
        #Create 2d array for the log of fbank features
        fbank_feat = logfbank(sig,rate)
        return(mfcc_feat)
```

Example 20

Project: *AudioEmotionDetection* Author: *DefinitelyNotTim* File: *emotionProcessor-threaded.py* MIT License 5 votes

```
def mfccProc2(self, results_dict):
        (rate,sig) = audioBasicIO.readAudioFile(self.fname)
        #Create 2d array for MFCC features
        mfcc_feat = mfcc(sig,samplerate = 44100, nfft = 1103)
        #Create 2d array for the delta of MFCC features
        d_mfcc_feat = delta(mfcc_feat, 2)
        #Create 2d array for the log of fbank features
        fbank_feat = logfbank(sig,rate)
        dev_array = []
        for i in mfcc_feat:
            temp = stdev(i)
            dev_array.append(temp)
        tone = stdev(dev_array)
        results_dict["tone"] = tone
        return(mfcc_feat)
```

Example 21

Project: *speaker-recognition-py3* Author: *crouchred* File: *features.py* Apache License 2.0 5 votes

```
def get_feature(fs, signal):
    mfcc_feature = mfcc(signal, fs)
    if len(mfcc_feature) == 0:
        print >> sys.stderr, "ERROR.. failed to extract mfcc feature:", len(signal)
    return mfcc_feature
```

Example 22

Project: *Speech-commands-recognition* Author: *lucko515* File: *audio_utils.py* MIT License 5 votes

```
def mfcc_pack(features):
    '''
    Packs all MFCC features into one single input matrix.

    :params:
        features - List of all MFCC features that are being packed. Example: [mfcc, delta1_mfcc]

    :returns:
        unified numpy matrix made of all MFCC features
    '''
    return np.hstack(features)
```

Example 23

Project: *persephone* Author: *persephone-tools* File: *feat_extract.py* GNU General Public License v3.0 5 votes

```
def extract_energy(rate, sig):
    """ Extracts the energy of frames. """

    mfcc = python_speech_features.mfcc(sig, rate, appendEnergy=True)
    energy_row_vec = mfcc[:, 0]
    energy_col_vec = energy_row_vec[:, np.newaxis]
    return energy_col_vec
```

Example 24

Project: *persephone* Author: *persephone-tools* File: *feat_extract.py* GNU General Public License v3.0 5 votes

```
def  mfcc(wav_path):
    """ Grabs MFCC features with energy and derivates. """

    (rate, sig) = wav.read(wav_path)
    feat = python_speech_features.mfcc(sig, rate, appendEnergy=True)
    delta_feat = python_speech_features.delta(feat, 2)
    all_feats = [feat, delta_feat]
    all_feats = np.array(all_feats)
    # Make time the first dimension for easy length normalization padding later.
    all_feats = np.swapaxes(all_feats, 0, 1)
    all_feats = np.swapaxes(all_feats, 1, 2)

    feat_fn = wav_path[:-3] + "mfcc13_d.npy"
    np.save(feat_fn, all_feats)
```

Example 25

Project: *mist-rnns* Author: *rdipietro* File: *timitphonemerec.py* Apache License 2.0 5 votes

```
def load(data_dir=DEFAULT_DATA_DIR, mfcc=True):
  """ Load all standardized TIMIT data with folded phoneme labels.

  Args:
    data_dir: A string. The data directory.
    mfcc: A boolean. If True, return MFCC sequences and their corresponding
      label sequences. Otherwise, return raw audio sequences in their
      associated label sequences.

  Returns:
    A tuple with 6 elements: train inputs, train labels, val inputs,
    val labels, test inputs, test labels. Each entry is a list of sequences.
    All input sequences are 2-D float arrays with shape
    `[length, values_per_step]` and all label sequences are 1-D int8 arrays
    with shape `[length]`.
  """
  types = ['mfcc', 'mfcc_labels'] if mfcc else ['audio', 'labels']
  ret = []
  for name in ['train', 'val', 'test']:
    for type in types:
      path = os.path.join(data_dir, name + '_' + type + '.npy')
      if not os.path.exists(path):
        raise ValueError('Data not found in %s. Run timit.py and timitphonemerec.py.' % data_dir)
      data = np.load(path)
      if type == 'audio':
        data = [seq[:, np.newaxis] for seq in data]
      ret.append(data)
  return tuple(ret)
```

Example 26

Project: *mist-rnns* Author: *rdipietro* File: *timitphonemerec.py* Apache License 2.0 5 votes 👍 👎

```
def load_split(data_dir=DEFAULT_DATA_DIR, val=True, mfcc=True, normalize=True):
  """ Load a standardized-TIMIT train, test split.

  Args:
    data_dir: A string. The data directory.
    val: A boolean. If True, return the validation set as the test set.
     mfcc: A boolean. If True, return MFCC sequences and their corresponding
      label Otherwise, return raw audio sequences in their associated
      label sequences.
    normalize: A boolean. If True, normalize each sequence individually by
      centering / scaling.

  Returns:
    A tuple, `(train_inputs, train_labels, test_inputs, test_labels)`. Each is
    a list of sequences. All inputs are 2-D float arrays with shape
    `[length, values_per_step]` and all labels are 1-D int8 arrays with shape
    `[length]`.
  """
  sequence_lists = load(data_dir=data_dir, mfcc=mfcc)
  train_inputs, train_labels, val_inputs, val_labels, test_inputs, test_labels = sequence_lists
  if val:
    test_inputs = val_inputs
    test_labels = val_labels
  if normalize:
    train_inputs = [seq - np.mean(seq, axis=0, keepdims=True) for seq in train_inputs]
    train_inputs = [seq / np.std(seq, axis=0, keepdims=True) for seq in train_inputs]
    test_inputs = [seq - np.mean(seq, axis=0, keepdims=True) for seq in test_inputs]
    test_inputs = [seq / np.std(seq, axis=0, keepdims=True) for seq in test_inputs]
  return train_inputs, train_labels, test_inputs, test_labels
```

Example 27

Project: *Speech_Signal_Processing_and_Classification* Author: *gionanide*
File: *mfcc_pca_feature.py* MIT License 5 votes 👍 👎

```
def mfcc_features_extraction(wav):
        inputWav,wav = readWavFile(wav)
        print inputWav
        rate,signal = wavv.read(inputWav)
        mfcc_features = mfcc(signal,rate)
        return mfcc_features,wav
```

Example 28

Project: *Speech_Signal_Processing_and_Classification* Author: *gionanide* File: *mfcc.py* MIT License 5 votes 👍 👎

```
def mfcc_features_extraction(wav):
        inputWav,wav = readWavFile(wav)
        rate,signal = wavv.read(inputWav)
        mfcc_features = mfcc(signal,rate)
        #n numpy array with size of the number of frames , each row has one feature vector
        return mfcc_features,wav
```

Example 29

Project: *Speech_Signal_Processing_and_Classification* Author: *gionanide* File: *mfcc.py* MIT License 5 votes 👍 👎

```
def mean_features(mfcc_features,wav):
        #make a numpy array with length the number of mfcc features
        mean_features=np.zeros(len(mfcc_features[0]))
        #for one input take the sum of all frames in a specific feature and divide them with the number of frames
        for x in range(len(mfcc_features)):
                for y in range(len(mfcc_features[x])):
                        mean_features[y]+=mfcc_features[x][y]
        mean_features = (mean_features / len(mfcc_features))
        print mean_features
        writeFeatures(mean_features,wav)
```

Example 30

Project: *spoken-digit-recognition* Author: *Ralireza* File: *main.py* GNU General Public License v3.0 5 votes 👍 👎

```
def feature_extractor(sound_path):
    sampling_freq, audio = wavfile.read(sound_path)
    mfcc_features = mfcc(audio, sampling_freq)

    return mfcc_features
```

Example 31

Project: *Voiceprint-Recognition* Author: *SunYanCN* File: *feature_extraction.py* Apache License 2.0 5 votes 👍 👎

```
def get_feature_vectors(file, directory, no_of_frames, start_frame):
    (rate,sig) = wav.read(os.path.join(directory, file))
    fbank_feat = logfbank(sig,rate,nfft=2048)
    mfcc_feat = mfcc(sig,rate,winlen=0.032,winstep=0.016,numcep=13,nfft=2048)

    d_mfcc_feat = delta(mfcc_feat, 2)
    dd_mfcc_feat = delta(d_mfcc_feat, 2)

    mfcc_vectors = mfcc_feat[start_frame:start_frame+no_of_frames,:no_of_features]
    dmfcc_vectors = d_mfcc_feat[start_frame:start_frame+no_of_frames,:no_of_features]
    ddmfcc_vectors = dd_mfcc_feat[start_frame:start_frame+no_of_frames,:no_of_features]
    fbank_vectors = fbank_feat[start_frame:start_frame+no_of_frames,:no_of_fbank_features]

    feature_vectors = numpy.hstack((mfcc_vectors, dmfcc_vectors, ddmfcc_vectors, fbank_vectors))
    return feature_vectors
```

Example 32

Project: *tensorflow-rnn-ctc* Author: *ugnelis* File: *utils.py* MIT License 5 votes 👍 👎

```
def read_audio_files(dir, extensions=['wav']):
    """
    Read audio files.

    Args:
        dir: string.
            Data directory.
        extensions: list of strings.
            File extensions.
    Returns:
        files: array of audios.
    """
    if not os.path.isdir(dir):
        logging.error("Audio files directory %s is not found.", dir)
        return None

    if not all(isinstance(extension, str) for extension in extensions):
        logging.error("Variable 'extensions' is not a list of strings.")
        return None

    # Get files list.
    files_paths_list = []
    for extension in extensions:
        file_glob = os.path.join(dir, '*.' + extension)
        files_paths_list.extend(glob.glob(file_glob))

    # Read files.
    files = []
    for file_path in files_paths_list:
        audio_rate, audio_data = wav.read(file_path)
        file = mfcc(audio_data, samplerate=audio_rate)
        files.append(file)

    files = np.array(files)
    return files
```

Example 33

Project: *feature_extraction* Author: *chenxinpeng* File: *prepro_extract_mfcc.py* MIT License 5 votes 👍 👎

```
def extract_video_wav(video_wavs_list, video_class):
    video_wav_path = 'data/audio_feats_mfcc/' + video_class
    if os.path.isdir(video_wav_path) is False:
        os.mkdir(video_wav_path)

    for idx, wav_path in enumerate(video_wavs_list):
        time_start = time.time()
        (rate, sig) = wav.read(wav_path)

        sig_c1 = sig[:, 0]
        sig_c2 = sig[:, 1]

        mfcc_feat_c1 = mfcc(sig_c1, rate)
        mfcc_feat_c2 = mfcc(sig_c2, rate)

        mfcc_feat = np.concatenate((mfcc_feat_c1, mfcc_feat_c2), axis=1)

        video_wav_mfcc_feat_save_path = os.path.join(video_wav_path, os.path.basename(wav_path).split('.')[0] + '.npy')

        if os.path.isfile(video_wav_mfcc_feat_save_path) is True:
            continue

        np.save(video_wav_mfcc_feat_save_path, mfcc_feat)

        print('{}  {}  time cost: {:.3f}'.format(idx, wav_path, time.time()-time_start))
```

Example 34

Project: *speech-to-text* Author: *kevobt* File: *processing.py* GNU General Public License v3.0 5 votes 👍 👎

```
def calculate_mfccs(audio_file_path):
    """
    For a given audio clip, calculate the corresponding feature

    :param audio_file_path:
    :return: A numpy array of size (NUMFRAMES by numcep) containing features. Each row holds 1 feature vector.
    """
    (rate, data) = wavfile.read(audio_file_path)
    return mfcc(data, rate, numcep=26)
```

Example 35

Project: *ZASR_tensorflow* Author: *Pelhans* File: *audio_featurizer.py* Apache License 2.0 5 votes 👍 👎

```
def _compute_specgram(self, samples, sample_rate):
        """Extract various audio features."""
        if self._specgram_type == 'linear':
            return self._compute_linear_specgram(
                samples, sample_rate, self._stride_ms, self._window_ms,
                self._max_freq)
        elif self._specgram_type == 'mfcc':
            return self._compute_mfcc(samples, sample_rate, self._stride_ms,
                                      self._window_ms, self._max_freq)
        else:
            raise ValueError("Unknown specgram_type %s. "
                             "Supported values: linear." % self._specgram_type)
```

Example 36

Project: *ZASR_tensorflow* Author: *Pelhans* File: *audio_featurizer.py* Apache License 2.0 5 votes 👍 👎

```
def _compute_mfcc(self,
                      samples,
                      sample_rate,
                      stride_ms=10.0,
                      window_ms=20.0,
                      max_freq=None):
        """Compute mfcc from samples."""
        if max_freq is None:
            max_freq = sample_rate / 2
        if max_freq > sample_rate / 2:
            raise ValueError("max_freq must not be greater than half of "
                             "sample rate.")
        if stride_ms > window_ms:
            raise ValueError("Stride size must not be greater than "
                             "window size.")
        # compute the 13 cepstral coefficients, and the first one is replaced
        # by log(frame energy)
        mfcc_feat = mfcc(
            signal=samples,
            samplerate=sample_rate,
            winlen=0.001 * window_ms,
            winstep=0.001 * stride_ms,
            highfreq=max_freq)
        # Deltas
        d_mfcc_feat = delta(mfcc_feat, 2)
        # Deltas-Deltas
        dd_mfcc_feat = delta(d_mfcc_feat, 2)
        # transpose
        mfcc_feat = np.transpose(mfcc_feat)
        d_mfcc_feat = np.transpose(d_mfcc_feat)
        dd_mfcc_feat = np.transpose(dd_mfcc_feat)
        # concat above three features
        concat_mfcc_feat = np.concatenate(
            (mfcc_feat, d_mfcc_feat, dd_mfcc_feat))
        return concat_mfcc_feat
```

Example 37

Project: *neuralmonkey* Author: *ufal* File: *speech.py* BSD 3-Clause "New" or "Revised" License 5 votes 👍 👎

```
def SpeechFeaturesPreprocessor(feature_type: str = "mfcc",
                               delta_order: int = 0,
                               delta_window: int = 2,
                               **kwargs) -> Callable:
    """Calculate speech features.

    First, the given type of features (e.g. MFCC) is computed using a window
    of length `winlen` and step `winstep`; for additional keyword arguments
    (specific to each feature type), see
    http://python-speech-features.readthedocs.io/. Then, delta features up to
    `delta_order` are added.

    By default, 13 MFCCs per frame are computed. To add delta and delta-delta
    features (resulting in 39 coefficients per frame), set `delta_order=2`.

    Arguments:
        feature_type: mfcc, fbank, logfbank or ssc (default is mfcc)
        delta_order: maximum order of the delta features (default is 0)
        delta_window: window size for delta features (default is 2)
        **kwargs: keyword arguments for the appropriate function from
            python_speech_features

    Returns:
        A numpy array of shape [num_frames, num_features].
    """

    if feature_type not in FEATURE_TYPES:
        raise ValueError(
            "Unknown speech feature type '{}'".format(feature_type))

    def preprocess(audio: Audio) -> np.ndarray:
        features = [FEATURE_TYPES[feature_type](
            audio.data, samplerate=audio.rate, **kwargs)]

        for _ in range(delta_order):
            features.append(delta(features[-1], delta_window))

        return np.concatenate(features, axis=1)

    return preprocess
```

Example 38

Project: *neuralmonkey* Author: *ufal* File: *speech.py* BSD 3-Clause "New" or "Revised" License 5 votes 👍 👎

```
def SpeechFeaturesPreprocessor(feature_type: str = "mfcc",
                               delta_order: int = 0,
                               delta_window: int = 2,
                               **kwargs) -> Callable:
    """Calculate speech features.

    First, the given type of features (e.g. MFCC) is computed using a window
    of length `winlen` and step `winstep`; for additional keyword arguments
    (specific to each feature type), see
    http://python-speech-features.readthedocs.io/. Then, delta features up to
    `delta_order` are added.

    By default, 13 MFCCs per frame are computed. To add delta and delta-delta
    features (resulting in 39 coefficients per frame), set `delta_order=2`.

    Arguments:
        feature_type: mfcc, fbank, logfbank or ssc (default is mfcc)
        delta_order: maximum order of the delta features (default is 0)
        delta_window: window size for delta features (default is 2)
        **kwargs: keyword arguments for the appropriate function from
            python_speech_features

    Returns:
        A numpy array of shape [num_frames, num_features].
    """

    if feature_type not in FEATURE_TYPES:
        raise ValueError(
            "Unknown speech feature type '{}'".format(feature_type))

    def preprocess(audio: Audio) -> np.ndarray:
        features = [FEATURE_TYPES[feature_type](
            audio.data, samplerate=audio.rate, **kwargs)]

        for _ in range(delta_order):
            features.append(delta(features[-1], delta_window))

        return np.concatenate(features, axis=1)

    return preprocess
```

Example 39

Project: *neuralmonkey* Author: *ufal* File: *speech.py* BSD 3-Clause "New" or "Revised" License 5 votes 👍 👎

```
def SpeechFeaturesPreprocessor(feature_type: str = "mfcc",
                               delta_order: int = 0,
                               delta_window: int = 2,
                               **kwargs) -> Callable:
    """Calculate speech features.

    First, the given type of features (e.g. MFCC) is computed using a window
    of length `winlen` and step `winstep`; for additional keyword arguments
    (specific to each feature type), see
    http://python-speech-features.readthedocs.io/. Then, delta features up to
    `delta_order` are added.

    By default, 13 MFCCs per frame are computed. To add delta and delta-delta
    features (resulting in 39 coefficients per frame), set `delta_order=2`.

    Arguments:
        feature_type: mfcc, fbank, logfbank or ssc (default is mfcc)
        delta_order: maximum order of the delta features (default is 0)
        delta_window: window size for delta features (default is 2)
        **kwargs: keyword arguments for the appropriate function from
            python_speech_features

    Returns:
        A numpy array of shape [num_frames, num_features].
    """

    if feature_type not in FEATURE_TYPES:
        raise ValueError(
            "Unknown speech feature type '{}'".format(feature_type))

    def preprocess(audio: Audio) -> np.ndarray:
        features = [FEATURE_TYPES[feature_type](
            audio.data, samplerate=audio.rate, **kwargs)]

        for _ in range(delta_order):
            features.append(delta(features[-1], delta_window))

        return np.concatenate(features, axis=1)

    return preprocess
```

Example 40

Project: *AcousticKeylogger* Author: *yossigor* File: *ModelGenerator.py* GNU General Public License v3.0 5 votes 👍 👎

```
def transform(X):
        return np.array([sf.mfcc(sample, 44100, 0.01, 0.0025, 32, 32, preemph=0, highfreq=12000, ceplifter=0,
                        appendEnergy=False).flatten() for sample in X])
```

Example 41

Project: *GlosBio* Author: *MikolajBalcerek* File: *mfcc.py* MIT License 5 votes 👍 👎

```
def calculate_mfcc_point(sample_rate, signal):
    """
    calculates mfcc features of single data point
    """
    # TODO: find better parameters
    return psf.mfcc(signal, samplerate=sample_rate, winlen=0.025, winstep=0.01, numcep=13,
             nfilt=26, nfft=512, lowfreq=0, highfreq=None, preemph=0.97,
             ceplifter=22, appendEnergy=True)
```

Example 42

Project: *GlosBio* Author: *MikolajBalcerek* File: *mfcc.py* MIT License 5 votes 👍 👎

```
def calculate_mfcc_list(sound_list):
    """
    uses the above function to calculate mfcc of list of data points
    """
    mfccs = []
    # TODO: maybe add other things like logfbang or deltas
    for (rate, signal) in sound_list:
        mfcc = calculate_mfcc_point(rate, signal)
        mfcc = preprocessing.scale(mfcc)
        mfccs.append(mfcc)
    return mfccs
```

Example 43

Project: *GlosBio* Author: *MikolajBalcerek* File: *mfcc.py* MIT License 5 votes 👍 👎

```
def calculate_mfcc_dict(data, verbose=False):
    """
    calculates mfcc features of the whole data kept in a dict
    that is returned from get_data in DataPreparation.load_data
    :param data the dictionary containing data:
    :param verbose True iff console output should be given:
    :return dict with features being mfcc vectors:
    """
    for id in data:
        if verbose:
            print("calculating mfcc of {}".format(id))
        data[id] = calculate_mfcc_list(data[id])
    return data
```

Example 44

Project: *GlosBio* Author: *MikolajBalcerek* File: *mfcc.py* MIT License 5 votes 👍 👎

```
def mfcc_dict_to_matrix(data):
    """
    transforms data contained in dict to np.array
    :param data a dictionary containing mfcc features:
    :return np.array of shape (num_data_points, num_features) and vector of labels:
    """
    X, y = [], []
    for id in data:
        for feat in data[id]:
            X.append(np.array(feat[1]))
            y.append(id)
    return np.stack(X, axis=1).T, np.array(y)
```

Example 45

Project: *GlosBio* Author: *MikolajBalcerek* File: *mfcc.py* MIT License 5 votes 👍 👎

```
def calculate_mfcc(data, verbose=False):
    """
    the main function of the module, that calculates matrices of mfcc features
    applies directly to result of get_data from DataPreparation.load_data

    """
    data = calculate_mfcc_dict(data, verbose)
    return mfcc_dict_to_matrix(data)
```

Example 46

Project: *GlosBio* Author: *MikolajBalcerek* File: *main.py* MIT License 5 votes 👍 👎

```
def update_plot_mfcc():
    new_file_name = file_menu.value
    data = load_data.load_raw_file(new_file_name)
    data_mfcc = np.swapaxes(psf.mfcc(data[1], data[0]), 0, 1)
    plot_mfcc.image(image=[data_mfcc], x=[0], y=[0], dw=[10], dh=[10])
```

Example 47

Project: *ASR_WORD* Author: *zw76859420* File: *file_wav.py* GNU Affero General Public License v3.0 5 votes 👍 👎

```
def get_mfcc_feature(wavsignal, fs):
    '''
    输入为wav文件数字表示和采样频率，输出为语音的MFCC特征+一阶差分+二阶差分；
    '''
    feat_mfcc = mfcc(wavsignal, fs)
    print(feat_mfcc)
    feat_mfcc_d = delta(feat_mfcc, 2)
    feat_mfcc_dd = delta(feat_mfcc_d, 2)
    wav_feature = np.column_stack((feat_mfcc, feat_mfcc_d, feat_mfcc_dd))
    return wav_feature
```

Example 48

Project: *Instrument-Classifier* Author: *seth814* File: *app.py* MIT License 5 votes 👍 👎

```
def init_model(self):
        yaml_file = open('mfcc.yaml', 'r')
        loaded_model_yaml = yaml_file.read()
        yaml_file.close()
        loaded_model = model_from_yaml(loaded_model_yaml)
        loaded_model.load_weights('mfcc.h5')
        loaded_model.compile(loss='categorical_crossentropy', optimizer='adam')
        self.model = loaded_model
        self.model._make_predict_function()
```

Example 49

Project: *Instrument-Classifier* Author: *seth814* File: *app.py* MIT License 5 votes 👍 👎

```
def init_figures(self):
        #time figure
        n_steps = 4
        step = 4410
        rate = 44100

        self.time_fig, ax = plt.subplots(1, figsize=(10, 5))
        x = np.arange(0, n_steps * step, 1)
        self.line, = ax.plot(x, np.random.rand(len(x)), '-', lw=2)

        half = int(np.power(2,16)/2)
        ax.set_title('Time Series')
        ax.set_xlabel('Length')
        ax.set_ylabel('Amplitude')
        ax.set_ylim(-half, half-1)
        ax.set_xlim(0, n_steps * step)
        plt.setp(ax, yticks=[])
        length = n_steps*step/rate
        self.t = np.arange(0, length, 1/rate)

        #mfcc figure
        self.mfcc_fig, self.mfcc_ax = plt.subplots(1, figsize=(10, 2))
        self.mfcc_ax.set_title('MFCC (64 filters)')
        X = np.zeros((9,64))
        self.mfcc_ax.imshow(X, cmap='hot', interpolation='nearest')

        #class figure
        self.class_fig, self.class_ax = plt.subplots(1, figsize=(1, 4))
        X = np.zeros((10,1))
        self.class_ax.imshow(X, cmap='hot', interpolation='nearest')
```

Example 50

Project: *Instrument-Classifier* Author: *seth814* File: *app.py* MIT License 5 votes

```
def build_kwargs(self):
        self.kwargs = {}
        self.kwargs['line'] = self.line
        self.kwargs['mfcc'] = self.mfcc_ax
        self.kwargs['class'] = self.class_ax
        self.kwargs['lock'] = self.lock
        self.kwargs['model'] = self.model
```