UNIVERSITY OF BUCHAREST

FACULTY OF

MATHEMATICS AND INFORMATICS

Artificial Intelligence

Dissertation Research Project

Speech emotion recognition based on audio recordings

Graduate:

Partu Ana-Maria

Scientific coordinator:

Conf. Dr. Paduraru Ciprian Ionut

Bucharest, June 2022

Table of contents

I. Introduction: Abstract and motivation

Emotion recognition is the field of machine learning where one artificial intelligence agent inferes through the use of various data points the emotion experienced by the human or animal in question. These data points can be categorized into images, videos, sounds, brain waves, other biological signals, etc.

With the increase in usage of social media and therefore the number of data points for this kind of classification, emotion recognition has been developed in recent years to a greater extent for multiple purposes. This field of study is getting more popular these days, having uses in areas such as psychology and customer need analysis, among others.

The aim of the project is to recognize emotions based on labeled audio recordings. This is done by extracting, cleaning and normalizing the sound waves in the recordings which are stored as sound files. The data extracted from the sound files comes in the form of signals which then get processed with various techniques to compute different coefficients and formats which represent the final features for the model. Data augmentation can be done at any point in the processing pipeline, either augmenting the sound or the extracted coefficients. The models used can vary from a wide range of neural networks. The purpose of the work put in this project is to compare the above models and techniques to one another to determine which is the best approach out of the ones presented in this paper.

I chose this topic because I have briefly worked with emotion recognition based on text and I was interested to explore how the development process and the results for this topic would apply to sound-based emotion recognition. Another point that raised my interest was the fact that I have also never worked with audio as input data, so this was an opportunity to learn new techniques and procedures and experiment on my own with different methods that I have never worked with before.

II. Theoretical concepts

2.1. Audio data and data augmentation

Audio data used in artificial intelligence is digital audio data and it comes in different formats. The most popular format for working with audio data is the .wav format as the data is not compressed. Sound is represented as a wave and it has few basic properties, frequency which is the number of waves made by the signal in a one unit of time, amplitude is the height of the signal and it represents the intensity of the sound and the speed of the sound is the distance traveled by a sound in one unit of time. In order to use the sound wave in any machine learning model it must be turned into an array of numbers. There are multiple tools that can read a .wav file and return a signal, the one used in this project is librosa. Librosa reads a .wav file and returns an array containing the signal and a number which represents the sample rate. The sample rate represents the frequency of the samples in the signal. [1]

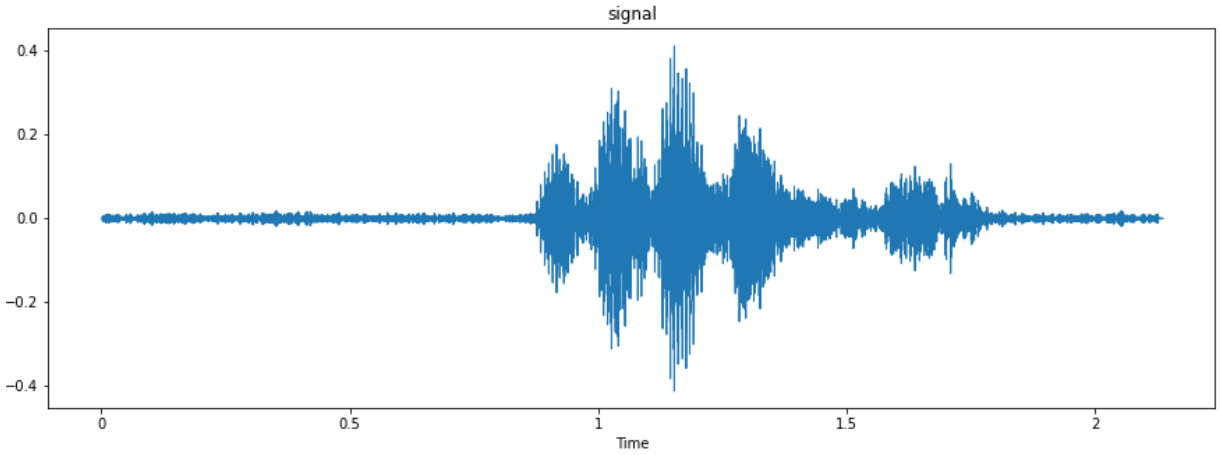

Fig. II.1.1. - Example of a signal

Audio data augmentation refers to the ways in which sound can be manipulated so that the sample is different but the contents are not significantly changed. When we talk about audio data augmentation there are two different versions of augmentation which can be applied on audio data: augmentation on raw audio (signal) or spectrogram augmentation. Noise addition, pitch shifting, time shifting and time stretching are all common augmentation methods for raw audio, while time and frequency masks are methods of augmentation for spectrograms.

Noise addition is a method of audio data augmentation which can be easily applied on a signal using numpy by simply adding some random value to the numbers in the signal array. Noise addition is useful in creating a model that is more robust and is able to recognize a class even in a noisy sample.

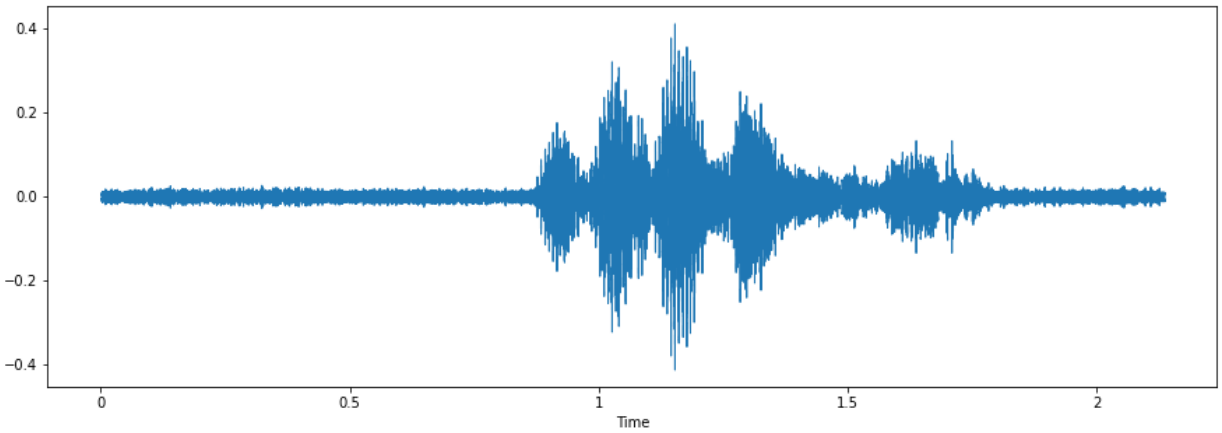

Fig. II.1.2. - Example of noise addition on a signal

The pitch of a sound represents how high or low a sound is. So pitch shifting moves the sound up or down without affecting the speed. Librosa has an effect which can change the pitch of the entire signal. [1]

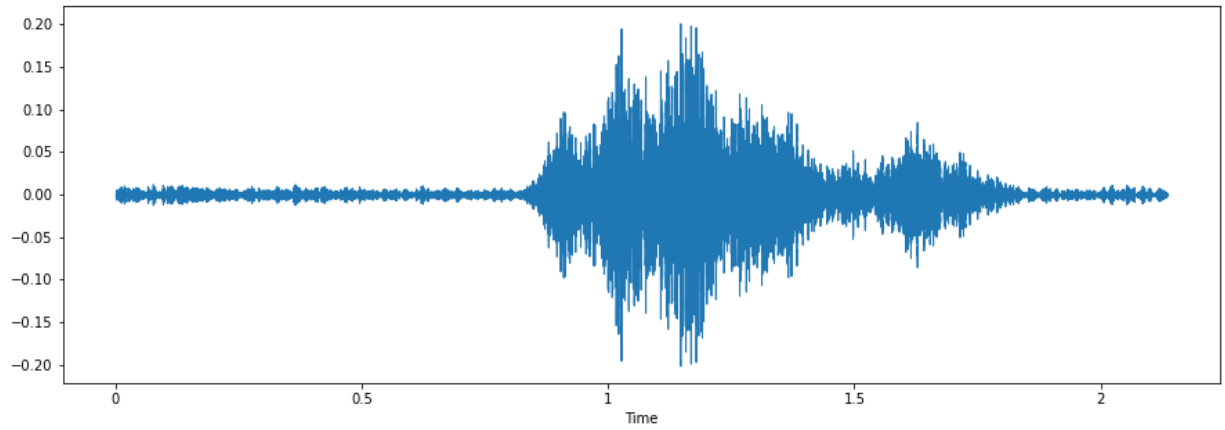

Fig. II.1.3 - Example of pitch shifting on a signal

Time shifting cuts a segment of the signal from a random point and moves it to a different part of the signal, creating a new sample.

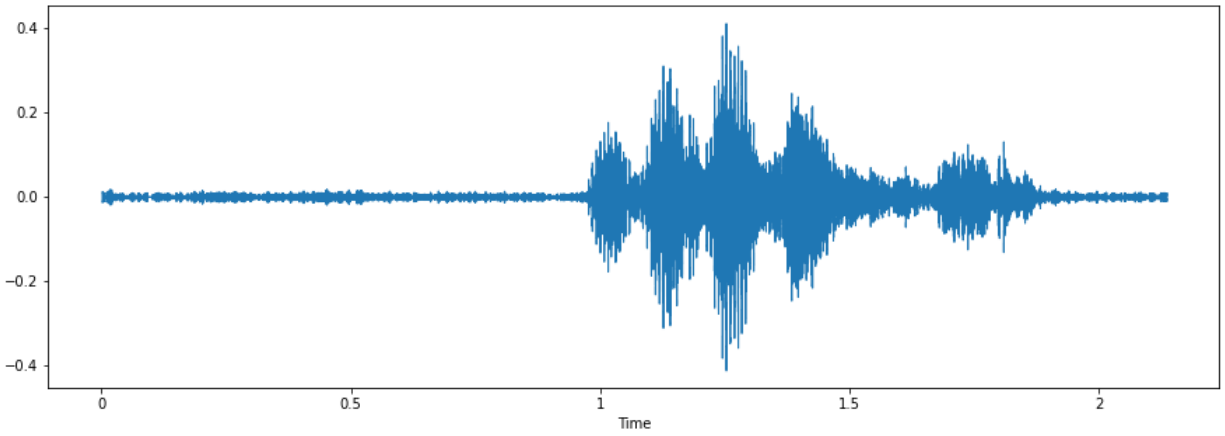

Fig. II.1.4. - Example of time shifting on a signal

Time stretching is a method that simply speeds up or slows down the sound without changing its pitch. Again, librosa has an effect that does this based on a speed factor that is given to the function.[1]

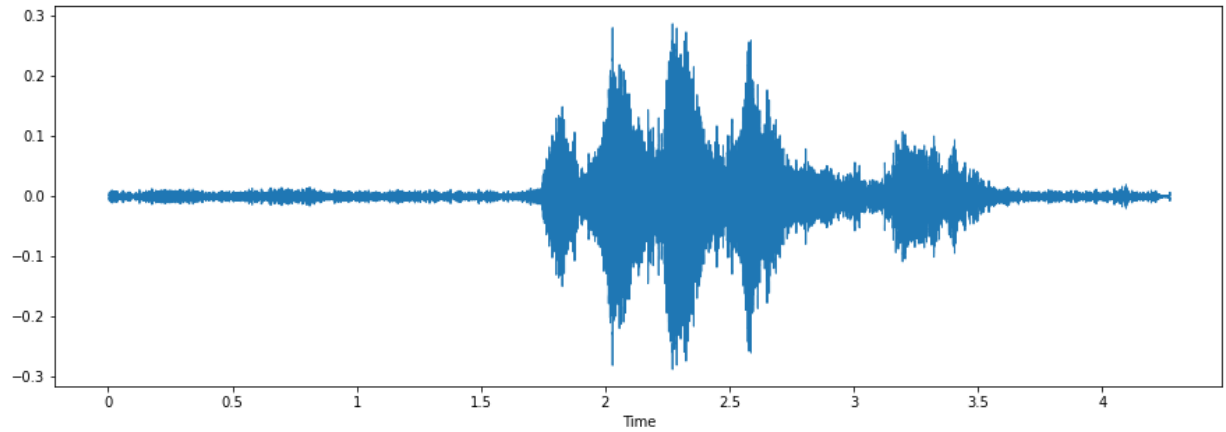

Fig II.1.5 - Example of time stretching to slow down a signal

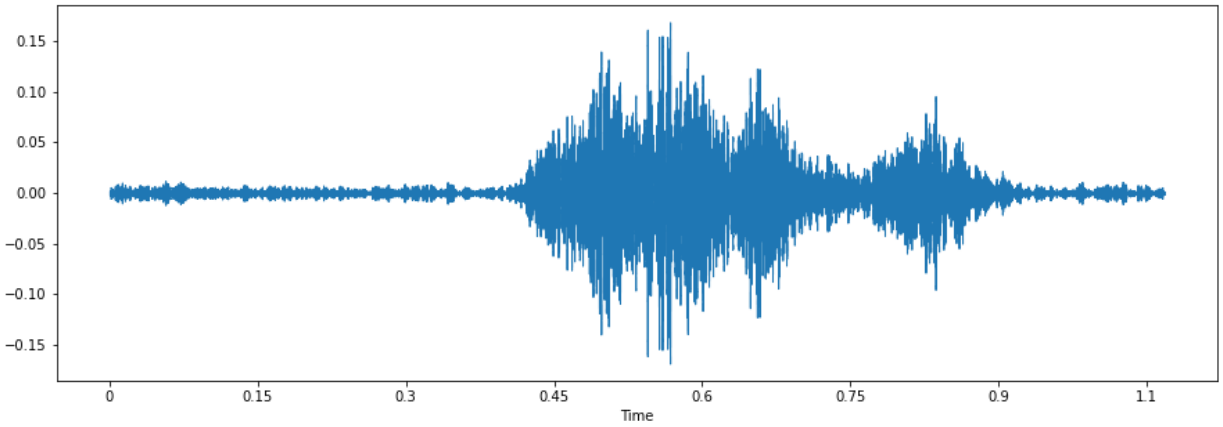

Fig. II.1.6 - Example of time stretching to speed up a signal

Time masking is a method of spectrogram augmentation that makes all the frequencies 0 at a particular time, thus hiding parts of the spectrogram on the time axis. [2][3]

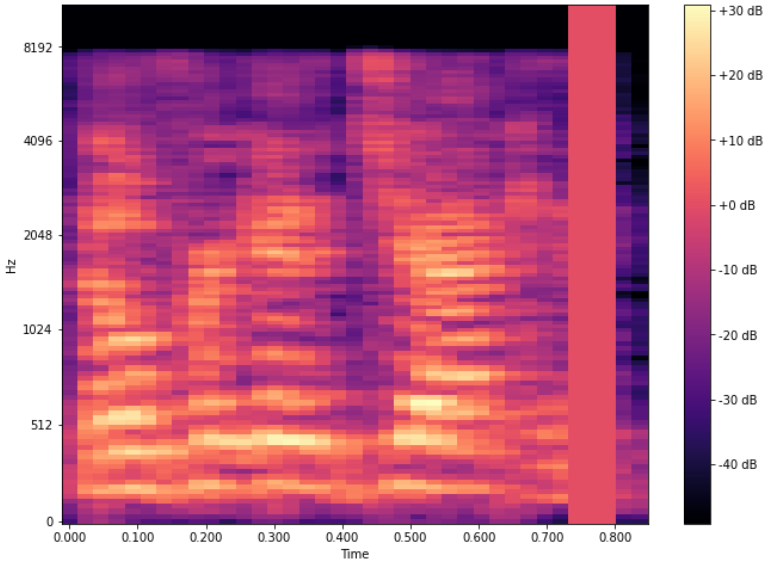

Fig. II.1.7 - Example of a time mask on a Mel Spectrogram

Frequency masking is a method of spectrogram augmentation that hides parts of the spectrogram on the frequency axis. By training on the whole sample as well as the same sample with “blind spots” the model can become more robust and be able to classify a sample even if it has imperfections.[2][3]

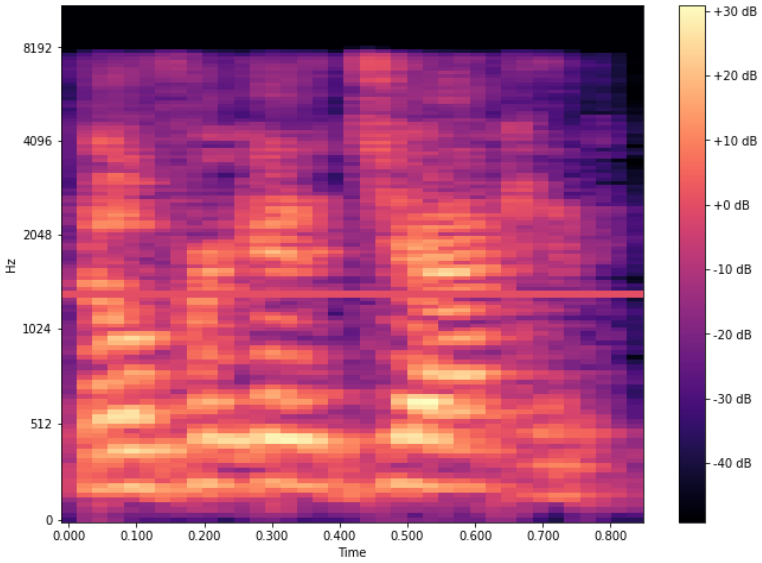

Fig. II.1.8 - Example of a frequency mask on a Mel Spectrogram

2.2. Feature extraction methods

A spectrogram visualizes the frequencies of a signal over time, in the form of a heat map. First, the Fast Fourier transformation is applied on overlapping windowed segments of the signal in order to create a spectrogram. The Mel spectrogram converts the frequencies to the Mel scale. Because it is a logarithmic scale for pitch, the perceptual distance between equal distances is the same.

$$\mathrm{Mel}(f) = 2595 \log\left(1 + \frac{f}{700}\right)$$

Fig. II.2.1 - Formula for converting Hertz to Mels [4]

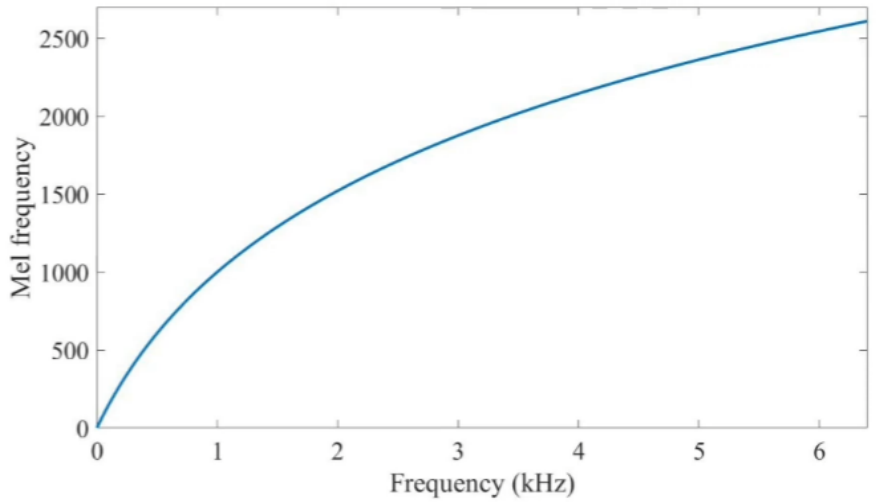

Fig. II.2.2 - Visualization of the Mel scale being a logarithmic scale

Librosa has a function that automatically computes the Mel Spectrogram of a given signal. In a normal spectrogram, frequencies are represented linearly while in a Mel Spectrogram, frequencies are represented logarithmically and humans perceive frequency logarithmically.

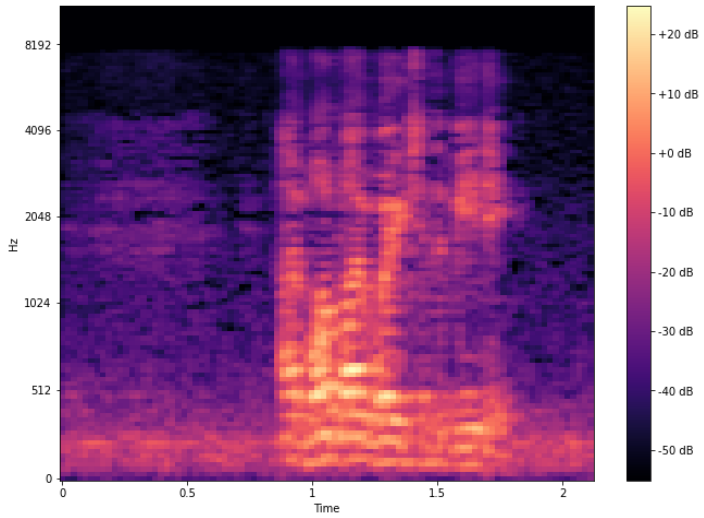

Fig. II.2.3 - Example of a Mel Spectrogram

The MFCC is a method of extracting features that give information about the shape of the spectral envelope. It is a compact representation of the spectrum. Librosa has a function which can extract the MFCC from a given signal. The number of coefficients computed can be

specified but when it comes to speech only the first 13 are relevant as they keep the information about formants and the spectral envelope. [5] [6][7]

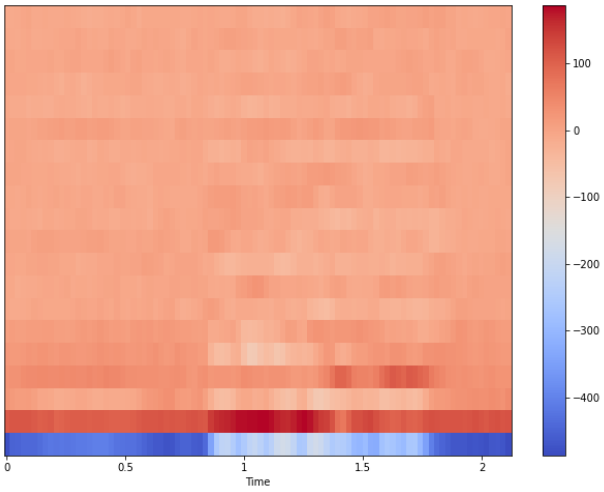

Fig. II.2.4 - Visualization of the MFCC

The delta and delta-delta of the MFCC represent the first two derivatives of the MFCC. The delta of MFCCs is calculated by subtracting the MFCCs of the previous frame from the current frame. The delta-delta of MFCCs is calculated by subtracting the delta of the previous frame from the delta of the current frame. [8]

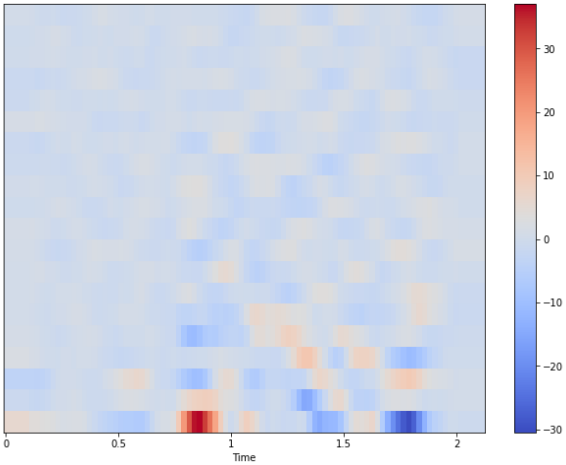

Fig. II.2.5 - Visualization of Delta

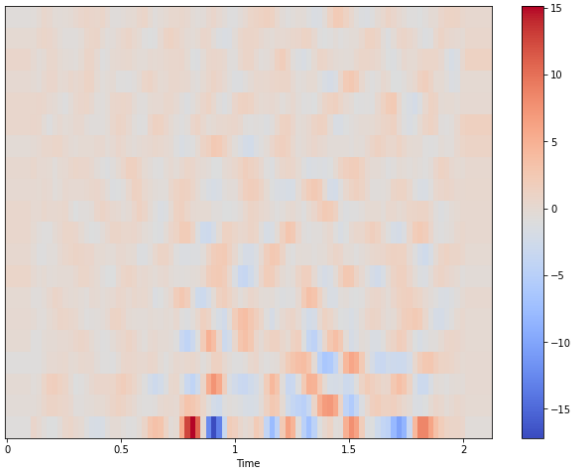

Fig. II.2.6 - Visualization of Delta-Delta

The root-mean-square represents the total magnitude of the signal or its loudness. Librosa has a function to automatically compute the RMS for a given signal.

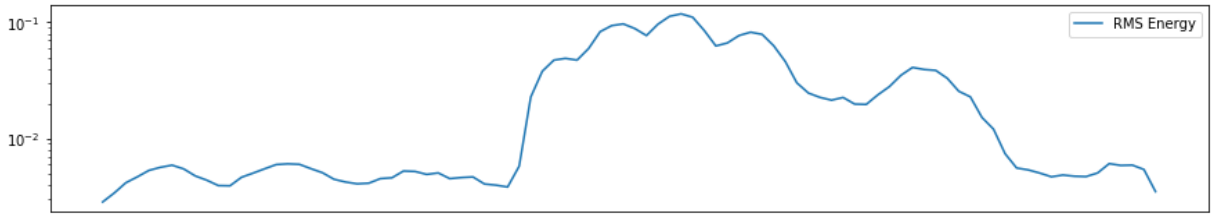

Fig. II.2.7 - Visualization of the RMS

The zero crossing rate represents the rate at which the signal moves from positive to negative through zero and the other way around. [9]

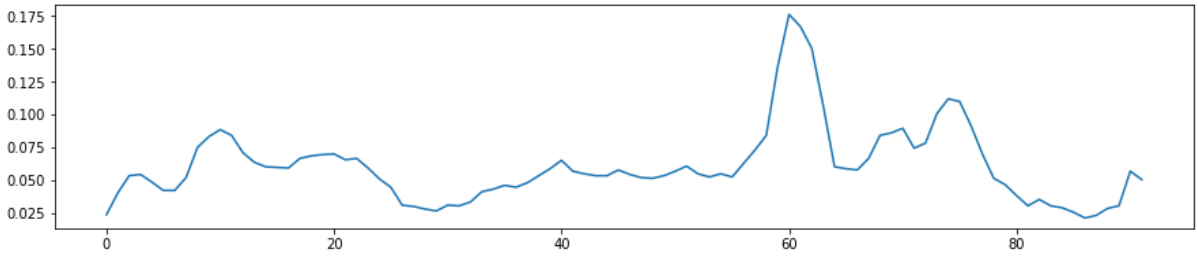

Fig. II.2.8 - Visualization of the Zero Crossing Rate

2.3. Artificial Intelligence and Machine Learning

The ability of a machine to perform tasks usually expected to be performed with human intelligence is called artificial intelligence. The goals of artificial intelligence are to learn, generalize, reason, and discover meaning.

One of the branches of artificial intelligence is machine learning. It is used to create software which predicts outcomes without being explicitly told what to do.

Neural networks are inspired by the human brain. They are algorithms designed to recognize patterns in a set of data, and they represent a subset of machine learning. Neural Networks contain multiple interconnected layers of neurons. The first layer is an input layer and it is followed by one or multiple hidden layers. The data travels through the hidden layers from the input layer and ends up in the last layer which is an output layer. Each neuron on each layer connects to others of another layer and it has a weight and a threshold. If the output value of a neuron is above the threshold, that value is sent to the next layer. [10]

2.3.1. CNNs

A CNN is an area of deep learning that specializes in pattern recognition. Each layer of a convolutional neural network receives an input, transforms the input and sends the output as an input for the next layer. Convolutional neural networks have three types of layers: convolutional layers which use a filter that performs the convolution operations and output a feature map, pooling layers are downsampling operations and fully connected layers which take the flattened output of the previous layer as input. In a fully connected layer, all neurons in one layer are connected to the ones in the next layer. [11][12]

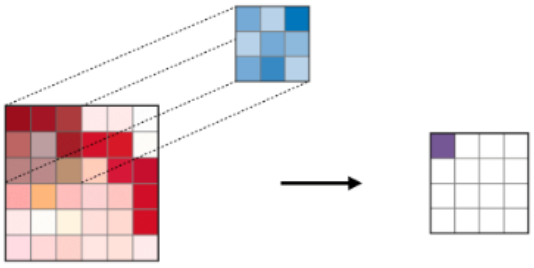

Fig. II.3.1.1.- Visualization of the process in a convolutional layer [11]

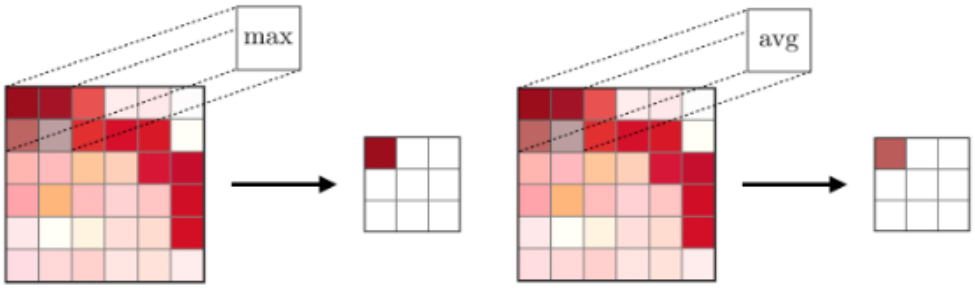

Fig. II.3.1.2.- Visualization of the process in the Max Pooling layer and the Average Pooling layer [11]

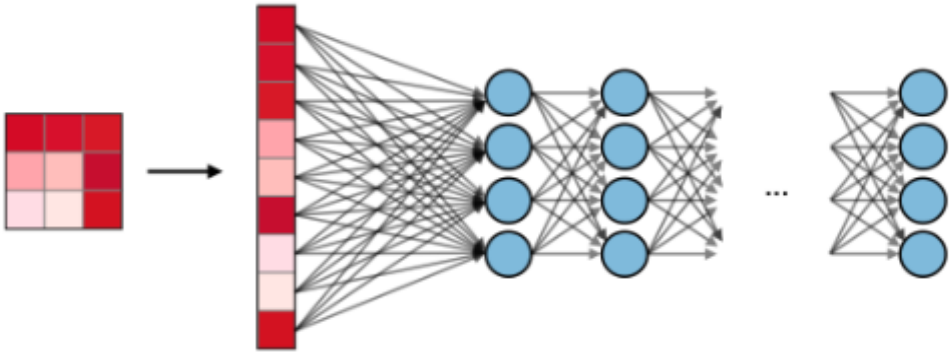

Fig. II.3.1.3.- Visualization of the process in the dense layers [11]

2.3.2. LSTM

Recurrent Neural Networks are neural networks which allow for the output of a layer to be used again as an input for that same layer along with the new input . The problem with RNNs is that they can only “remember” a certain amount of steps behind so they eventually lose context. Long Short Term Memory or LSTM that does not have the same problem as a normal RNN, as it is able to filter the most important parts and “remember” only those for more steps. So it can keep context but “forget” the information that is no longer applicable. The LSTM adds a state that is received by the layer; this state is called a LSTM cell. A LSTM cell has three parts, an input gate, a forget gate,and an output gate. The input gate serves the purpose of telling the layer which information should be added while the forget gate tells the layer which information is no longer

applicable and can be “forgotten”.The output gate tells the layer which part of all the information stored in the state should be the output. [13]

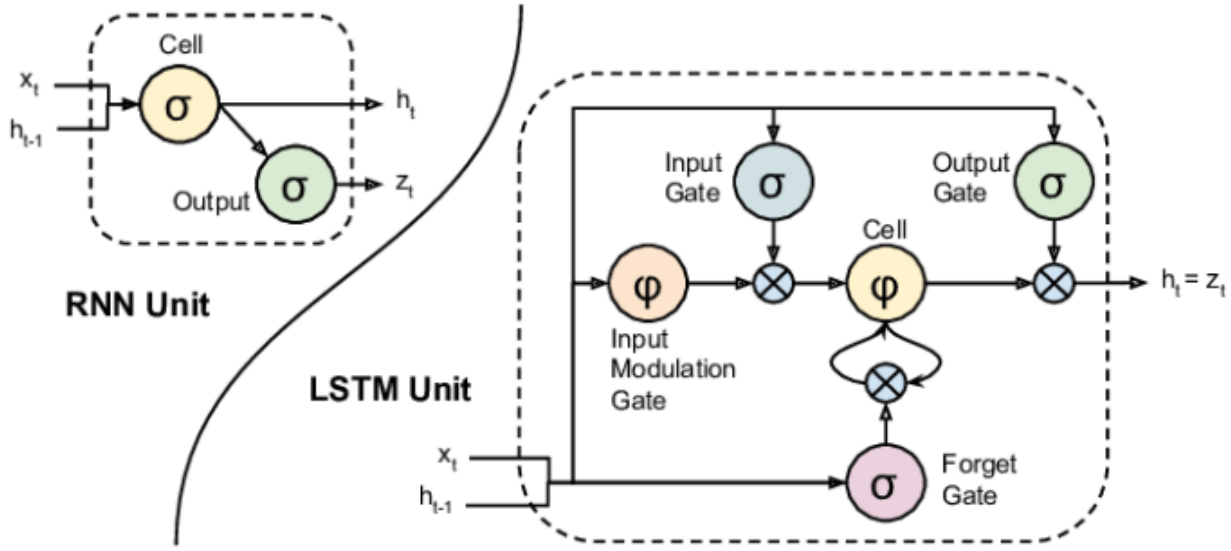

Fig. II.3.2.1. - Visualization of a RNN cell and a LSTM cell

III. Dataset

The main dataset chosen for the purposes of this project is the CREMA-D ("Crowd-sourced Emotional Multimodal Actors Dataset") dataset. It is a crowd-sourced audio-visual dataset created for emotion recognition. It contains around 7400 clips of 91 actors speaking from a collection of 12 sentences. Each of the sentences are spoken by each actor with different emotions. The displayed emotions are anger, disgust, fear, happiness, sadness and neutral. [14]

On top of the CREMA-D dataset, others were added in order to add more samples and variety to the dataset. The second dataset added is RAVDESS ("Ryerson Audio-Visual Database of Emotional Speech and Song"). It contains around 1400 audio-only files of 24 actors uttering two statements with different emotions. The displayed emotions are anger, disgust, fear, happiness, sadness, neutral, calmness and surprise. Both the CREMA-D and the RAVDESS datasets contain both male and female voices. [15]

The TESS ("Toronto Emotional Speech Set") dataset is the third dataset added and it contains around 2800 audio recordings of phrases spoken by two actresses displaying different emotions. The displayed emotions are anger, disgust, fear, happiness, sadness, neutral and surprise. [16]

The fourth and final dataset added is SAVEE ("Surrey Audio-visual Expressed Emotion"). It contains around 480 audio files of recorded speech by 4 different male actors displaying different emotions. The displayed emotions are anger, disgust, fear, happiness, sadness, surprise and an unknown class. [17]

For the purpose of this project 4 different datasets were created out of the 4 described above. The first dataset contains all of the samples combined and is labeled with the 7 predominant emotions: anger, fear, happiness, sadness, disgust, neutral and surprise. The calmness and unknown labels and samples were dropped as the number of samples for them was very low in comparison with the rest.

The second and third dataset contain all the male and female samples separately and are labeled with the 7 main emotions.

The forth dataset contains all of the samples combined but it is labeled with both the sex and the emotion (eg. instead of "anger" there are now two labels: "male_anger" and "female_anger"), thus doubling the number of labels to a total of 14.

The dataset were created for the sake of comparison, to see if sex played a role in choosing the feature extraction method or the model and to check if some models might perform better with a higher number of labels and a more in depth separation of the classes.

The labels for each sample were extracted separately from each original dataset since they all had different naming for the .wav files and the CREMA-D dataset had a separate .csv file explaining the naming method of the files. All the labels were matched when they were extracted (eg. some had “happy” as a label, some had “happiness” as a label so all the “happy” labels were saved as “happiness”).

It should be mentioned that all of the samples were recorded by actors so the display of the emotion is not natural.

3.1. Exploratory Data Analysis and Information Visualization

Before processing could be done, some exploratory data analysis was necessary. The first extracted information was the gender distribution of the combined datasets.

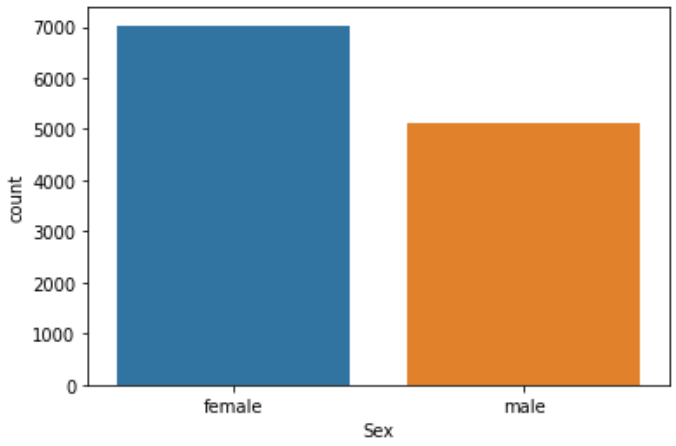

Fig. III.1.1. - Visualization of the gender distribution of the samples

The next step was extracting the class distribution over all the samples.

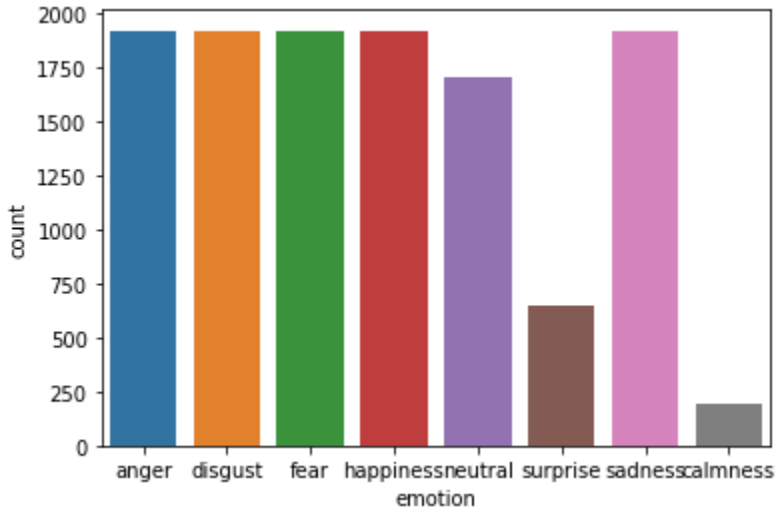

Fig. III.1.2. - Class distribution of all the samples

After seeing the class distribution, some samples and one class were removed due to a very small number of samples in comparison with the others. The removed class was “calmness”.

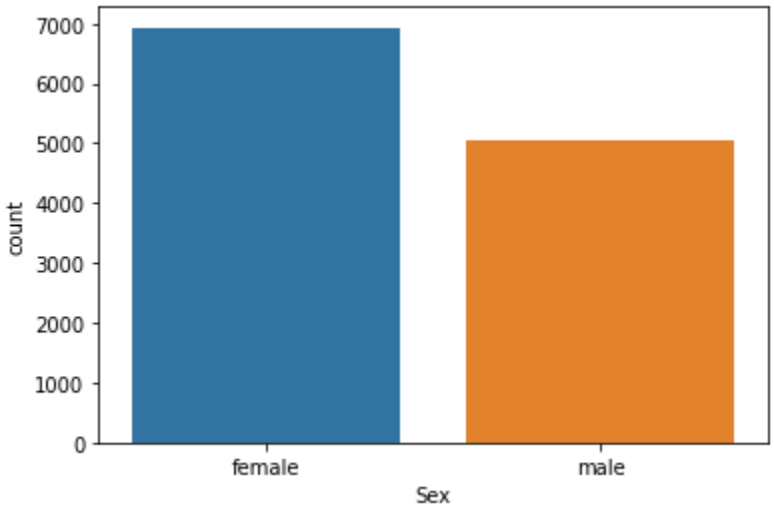

Fig. III.1.3. - Gender distribution of samples after removing a class and some corrupted files

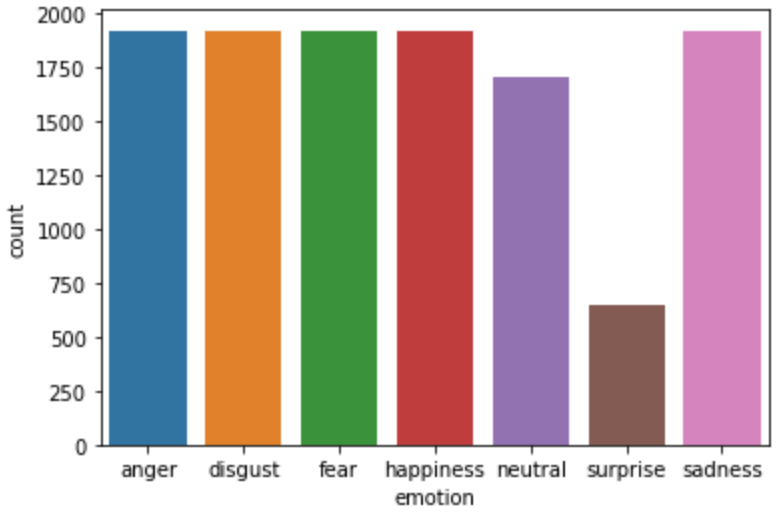

Fig. III.1.4. - Visualization of the class distribution of all the samples in the All samples 7 set

Fig. III.1.5. - Visualization of the class distribution of the samples in the Male samples set

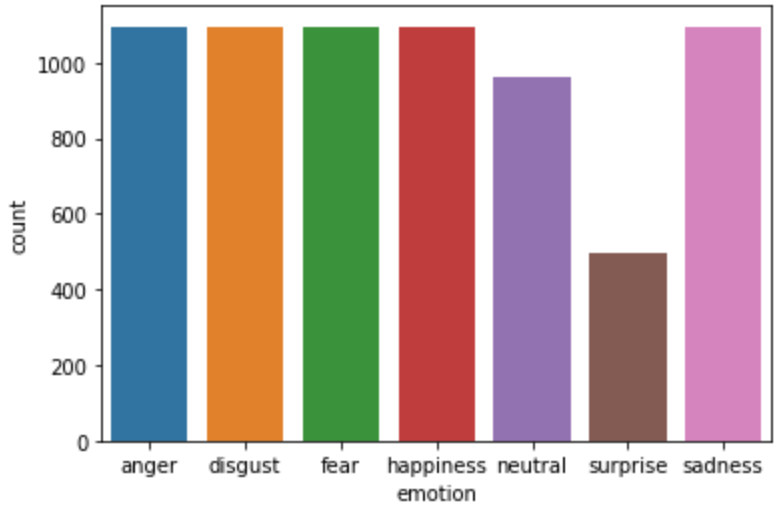

Fig. III.1.6. - Visualization of the class distribution of the samples in the Female samples set

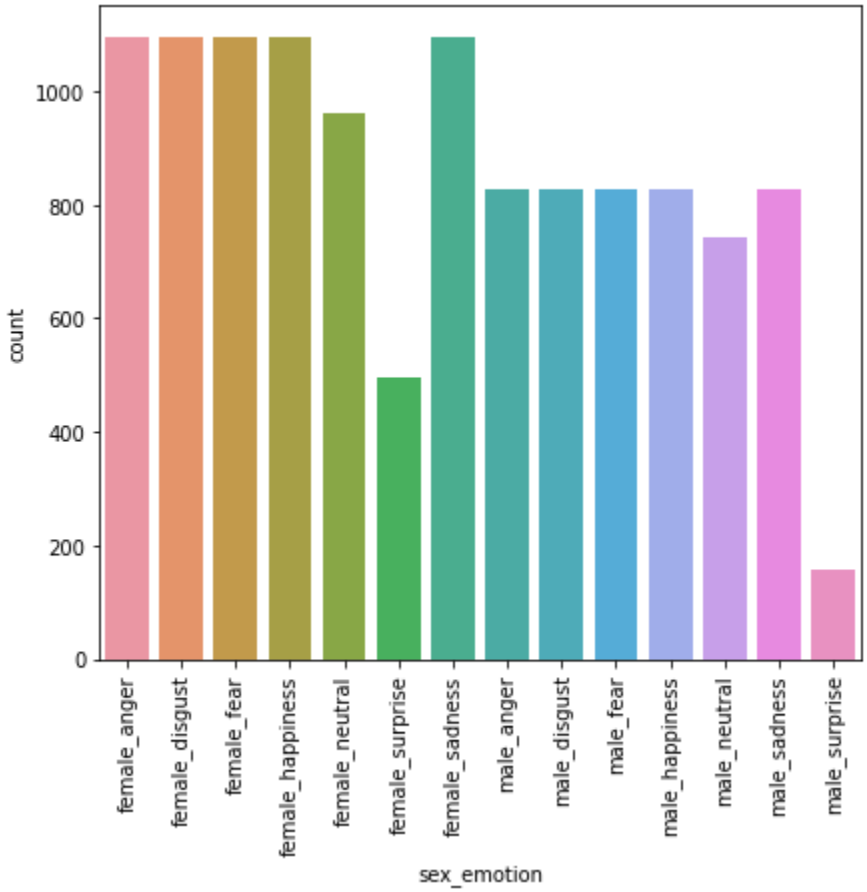

Fig. III.1.7. - Visualization of the class distribution of samples in the All samples 14 set

Once the final set of samples was decided, visualizing some signal from each of the original datasets was necessary in order to decide what processing needs to be done on the samples in order for them to be in the same format.

FIGS of signals from each original dataset

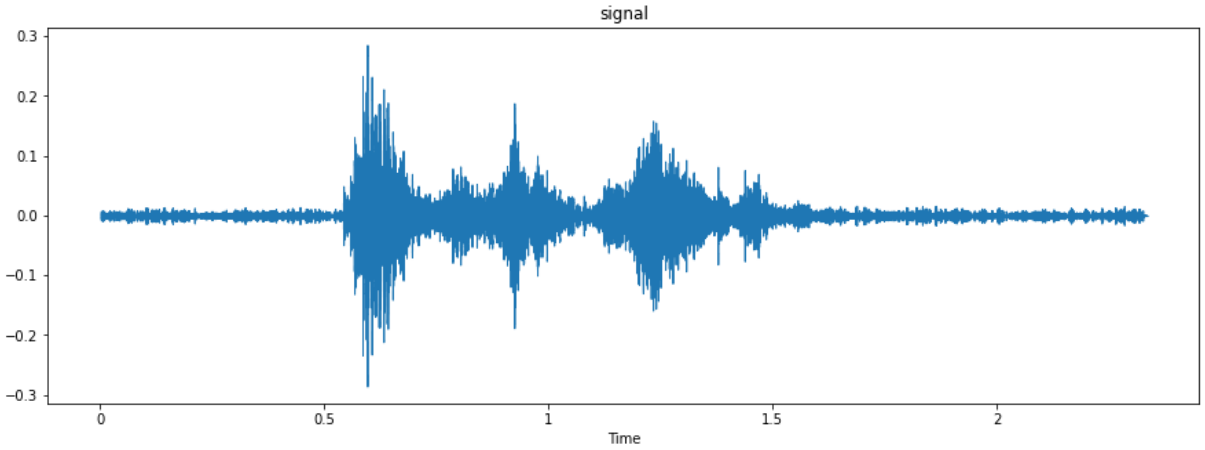

Fig. III.1.8. - Example signal from CREMA dataset

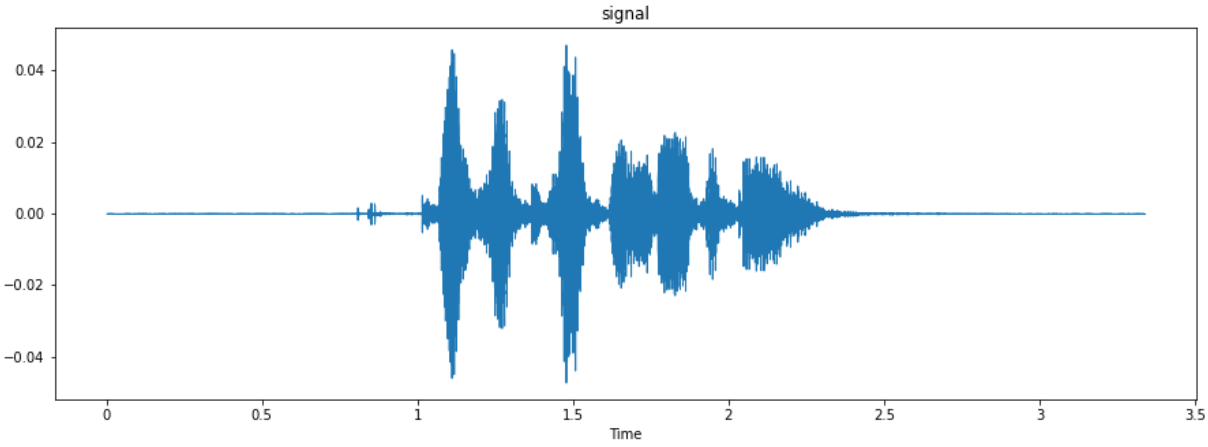

Fig. III.1.9. - Example signal from RAVDESS dataset

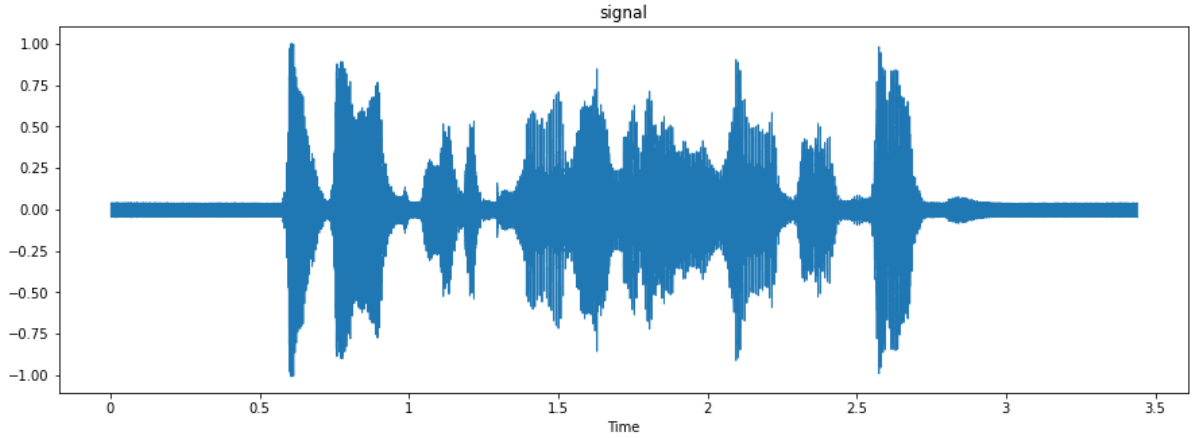

Fig. III.1.10. - Example signal from SAVEE dataset

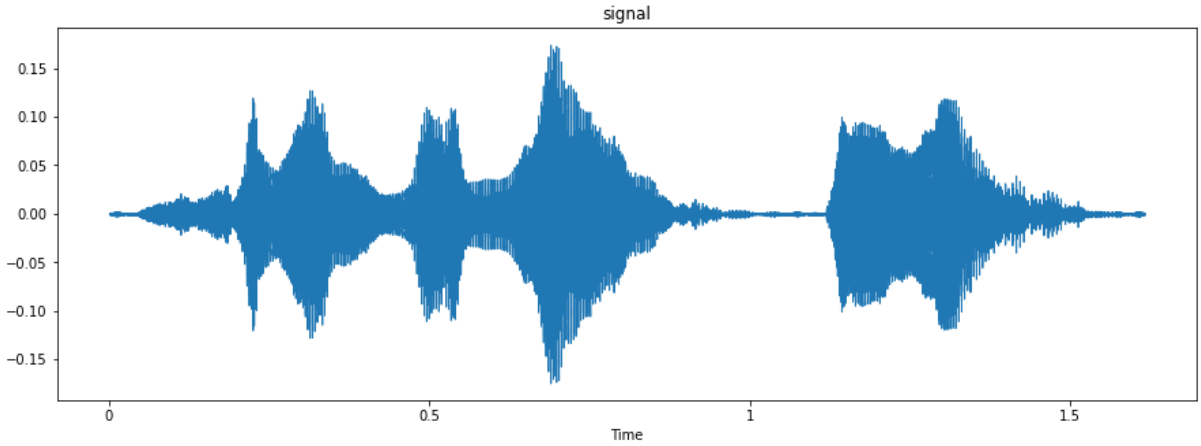

Fig. III.1.11. - Example signal from TESS dataset

All of the samples were then normalized using librosa. The normalized signal is vizualised in the figure below.

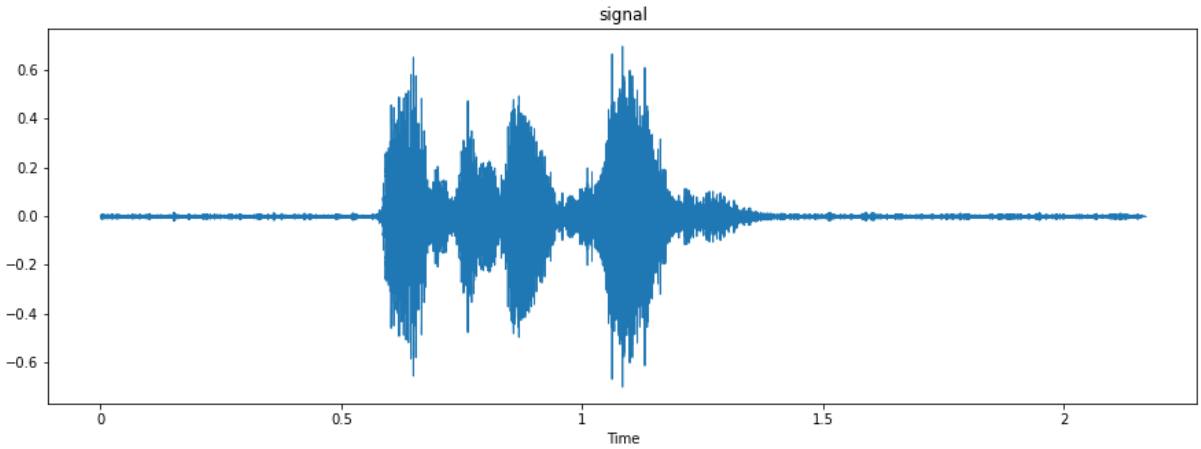

Fig. III.1.12 - Example of signal before preprocessing

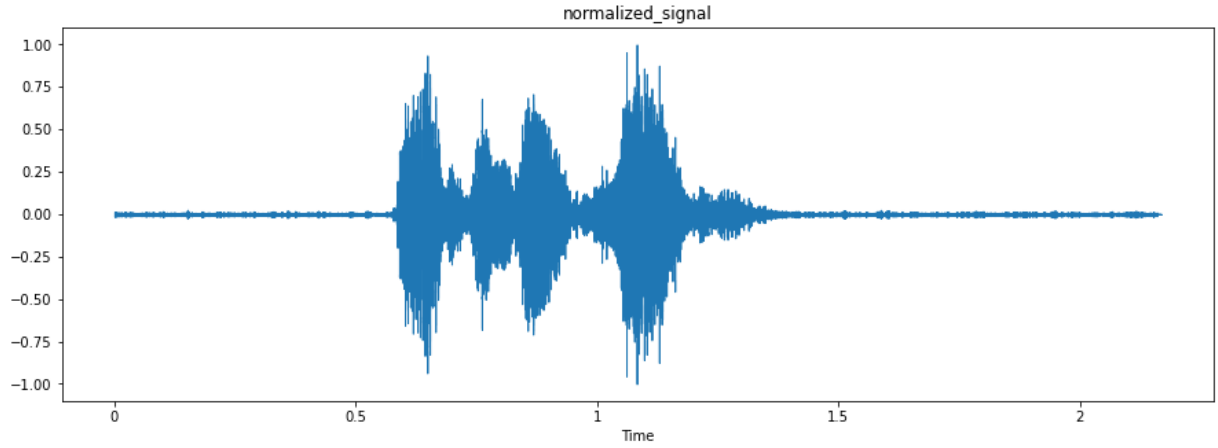

Fig. III.1.13 - Example of signal after normalization

The signals were then trimmed in order to get rid of the silence at the start and at the end of each file.

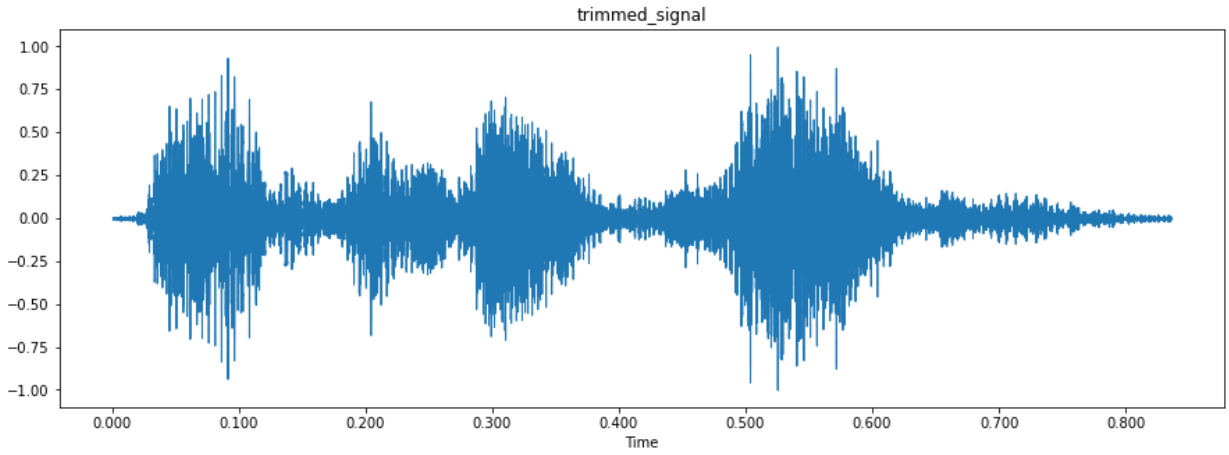

Fig. III.1.14 - Example of signal after trimming

In order for the data to be input into a model they need to be the same shape, to make sure the data is all in the same shape all of the signals need to be padded at the end. In order to know at which length the signals needed to be padded, the max length of the trimmed signals was calculated and used in the padding function.

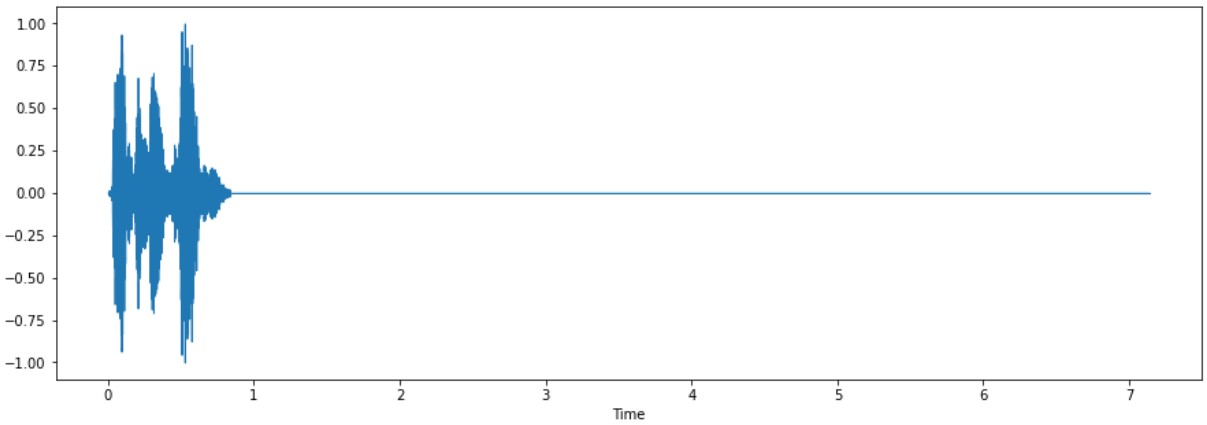

Fig. III.1.15 - Example of signal after padding

A range of data augmentation methods that could be used was visualized on an example signal. The methods tried are: noise addition, pitch shifting, time shifting, time stretching to slow down the signal and time stretching to make the signal faster.

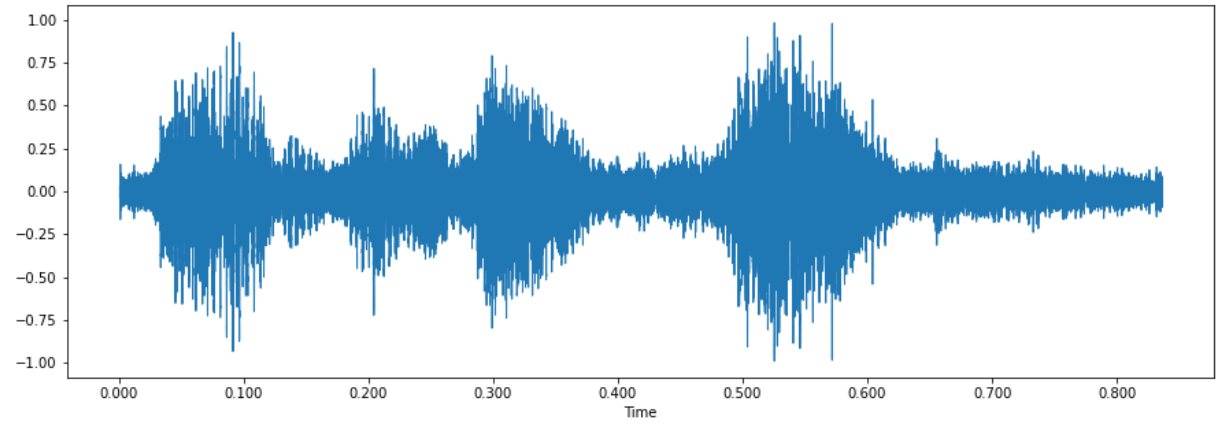

Fig. III.1.16 - Example of signal after noise addition

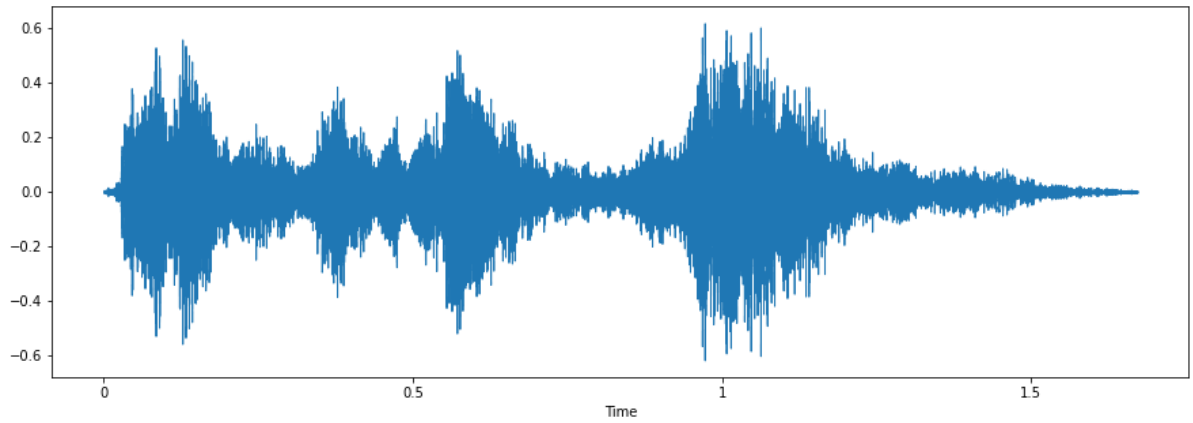

Fig. III.1.17 - Example of signal after time stretching the signal to be slower

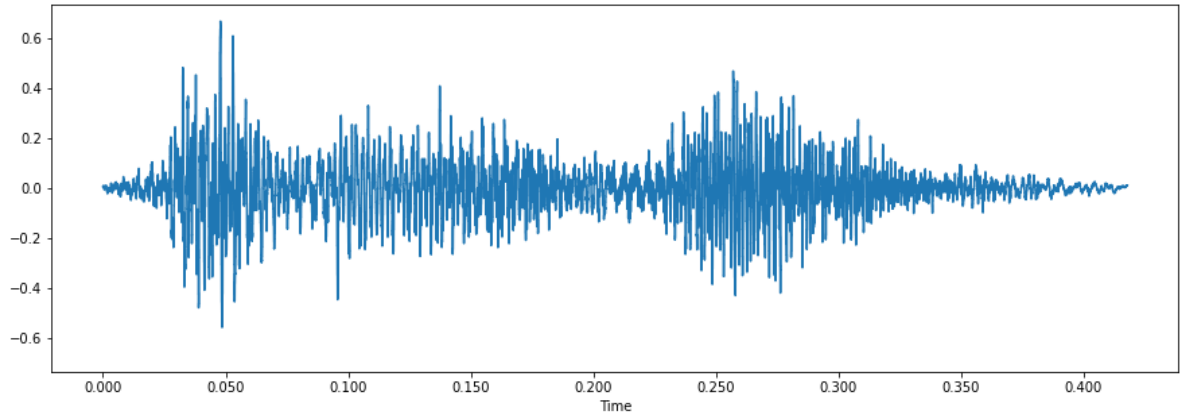

Fig. III.1.18 - Example of signal after time stretching the signal to be faster

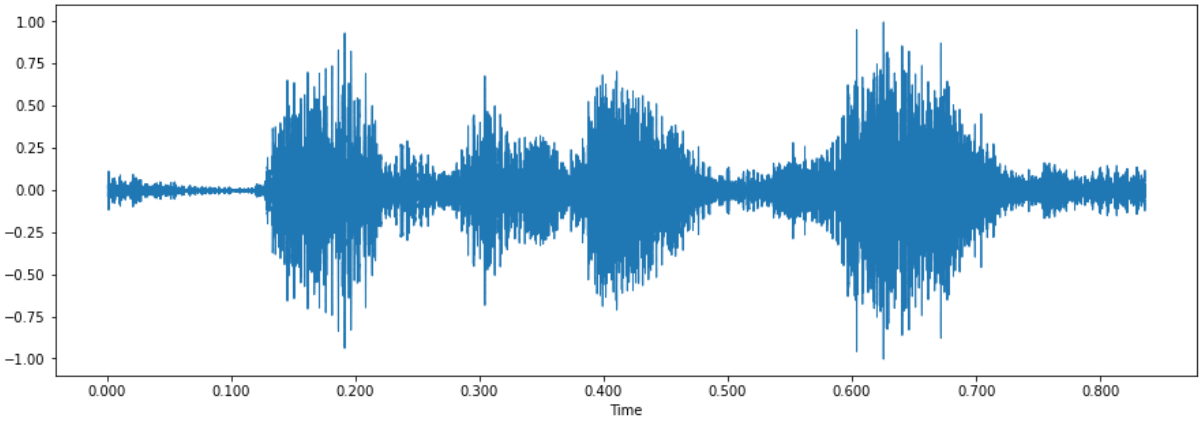

Fig. III.1.19 - Example of time shifted signal

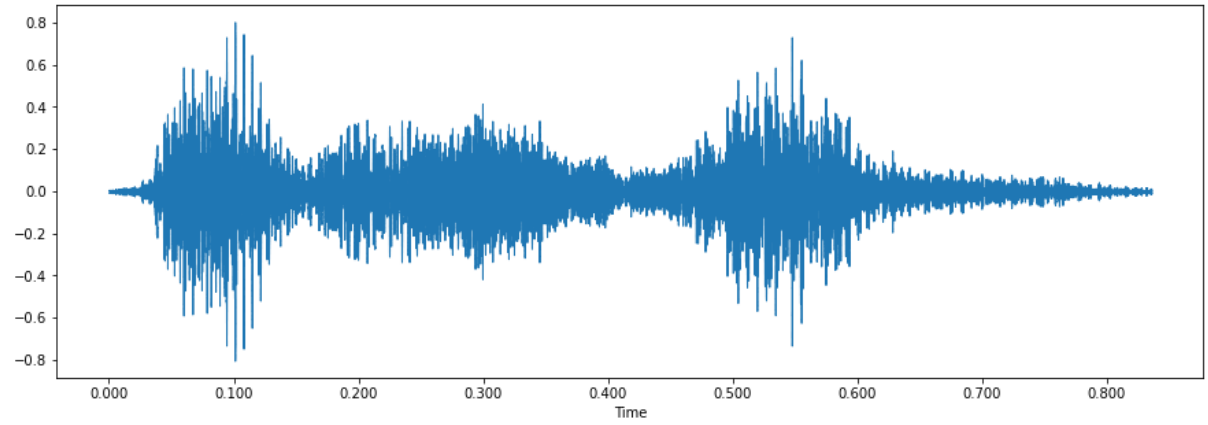

Fig. III.1.20 - Example of pitch shifted signal

Next, the feature extraction methods that were going to be tried were visualized. First the mel spectrogram was extracted from the signal. Next, the MFCC and Deltas were extracted.

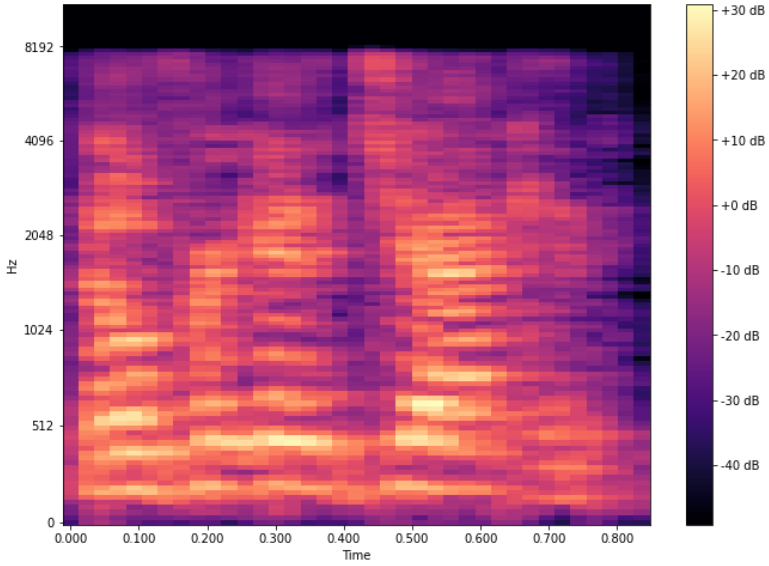

Fig. III.1.21. - Example of Mel Spectrogram

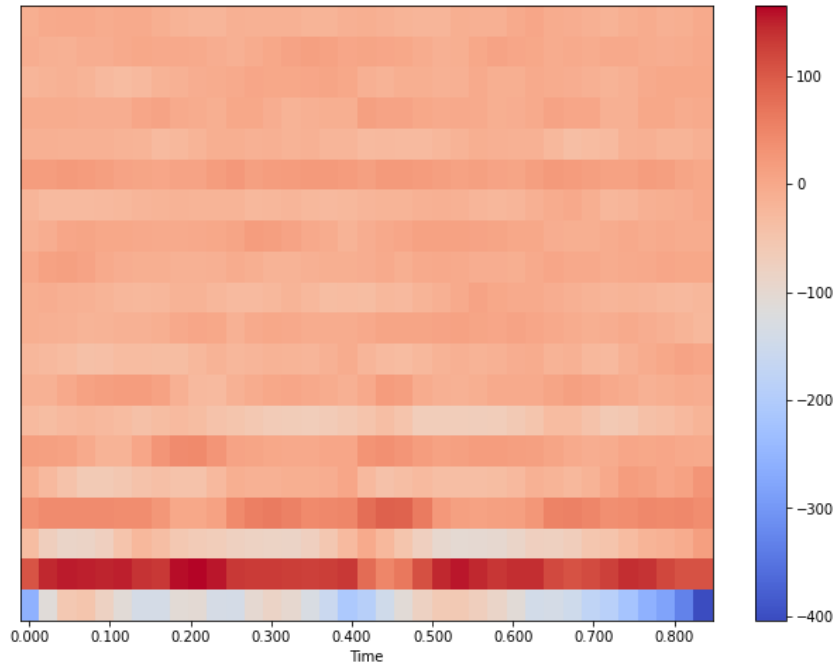

Fig. III.1.22. - Example of MFCC

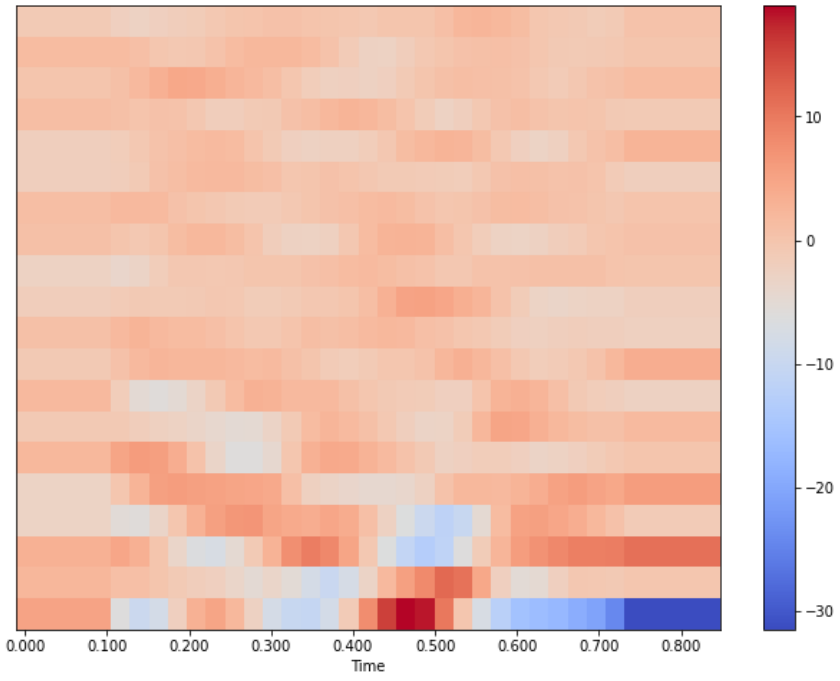

Fig. III.1.23. - Example of Delta

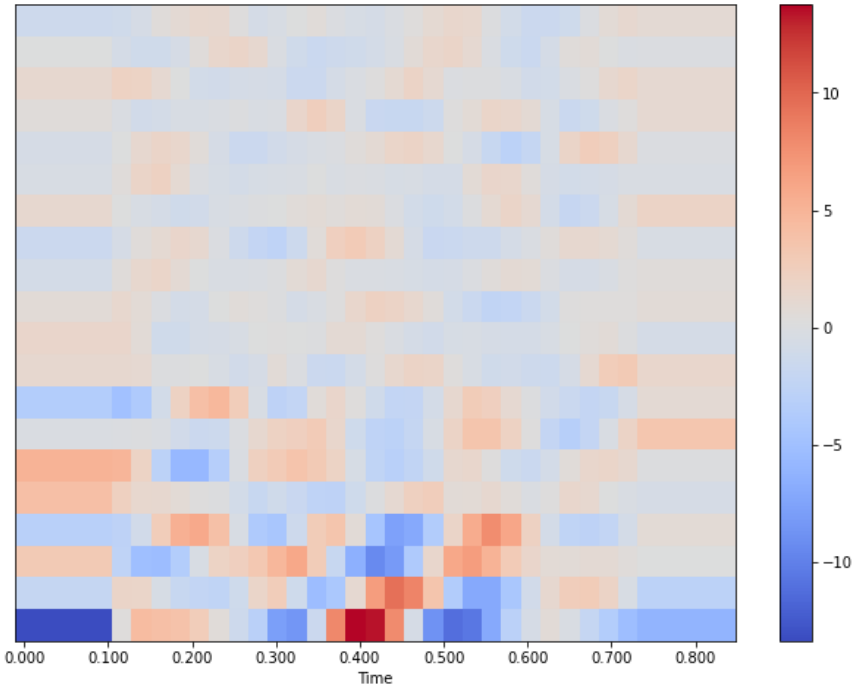

Fig. III.1.24. - Example of Delta-Delta

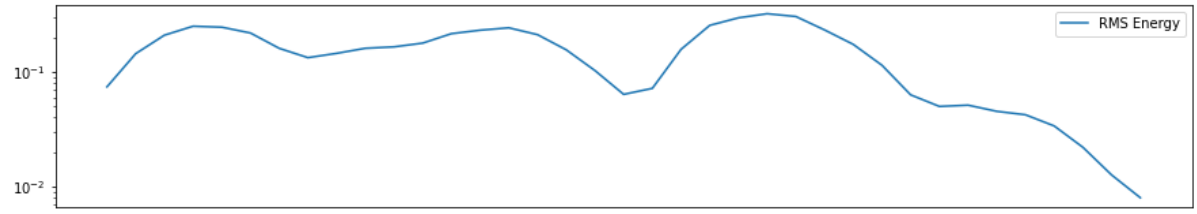

Fig. III.1.25. - Example of RMS

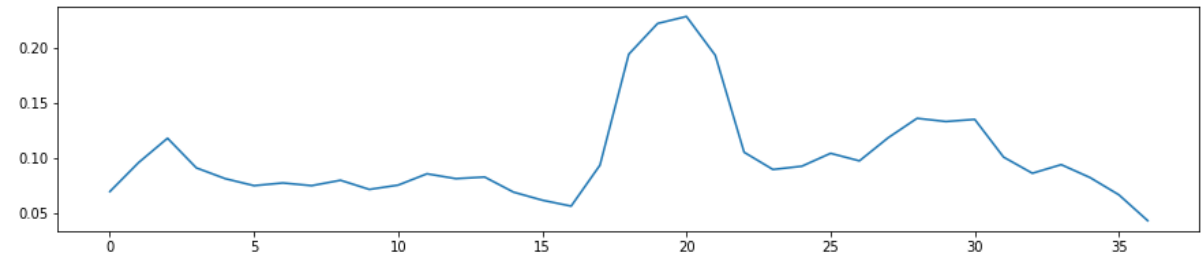

Fig. III.1.26. - Example of Zero Crossing Rate

A second method of data augmentation was considered, masking the spectrograms. A frequency mask and a time mask were visualized.

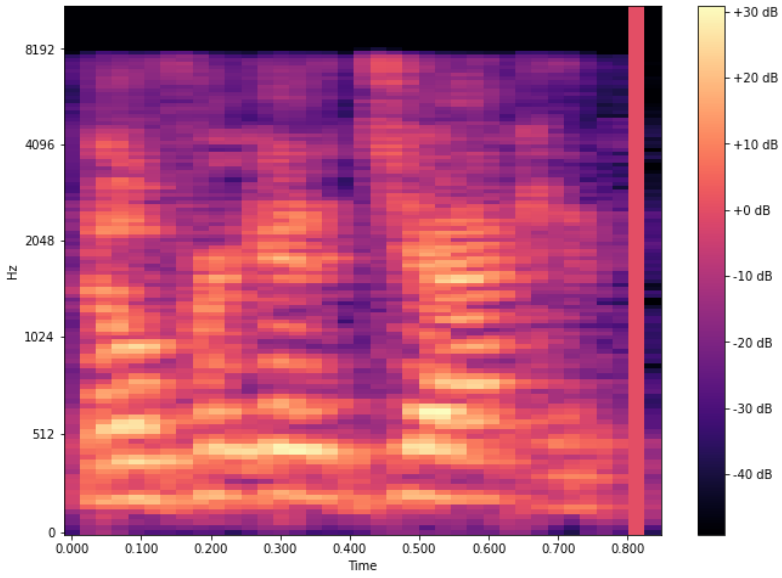

Fig. III.1.27. - Example of Mel Spectrogram with a time mask applied

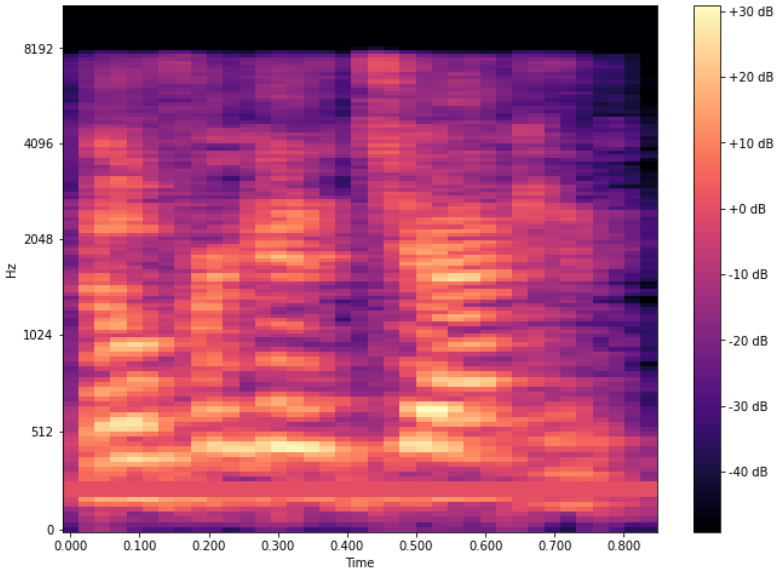

Fig. III.1.28. - Example of Mel Spectrogram with a frequency mask applied

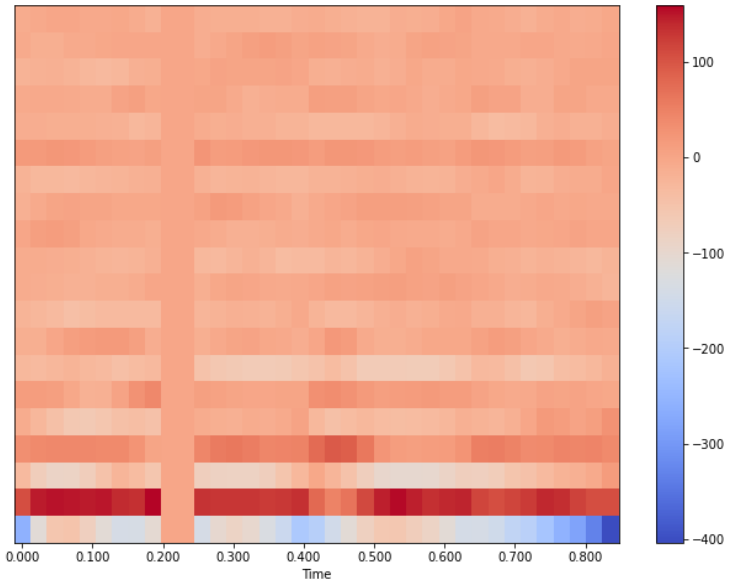

Fig. III.1.29. - Example of MFCC with a time mask applied

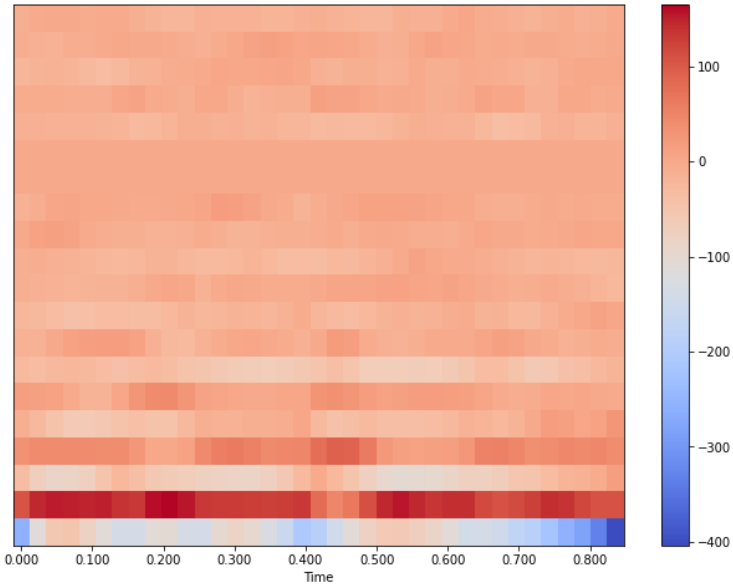

Fig. III.1.30. - Example of MFCC with a frequency mask applied

The signal, mel spectrogram and mfcc were visualized for each class in order to see how much they differed to the naked eye.

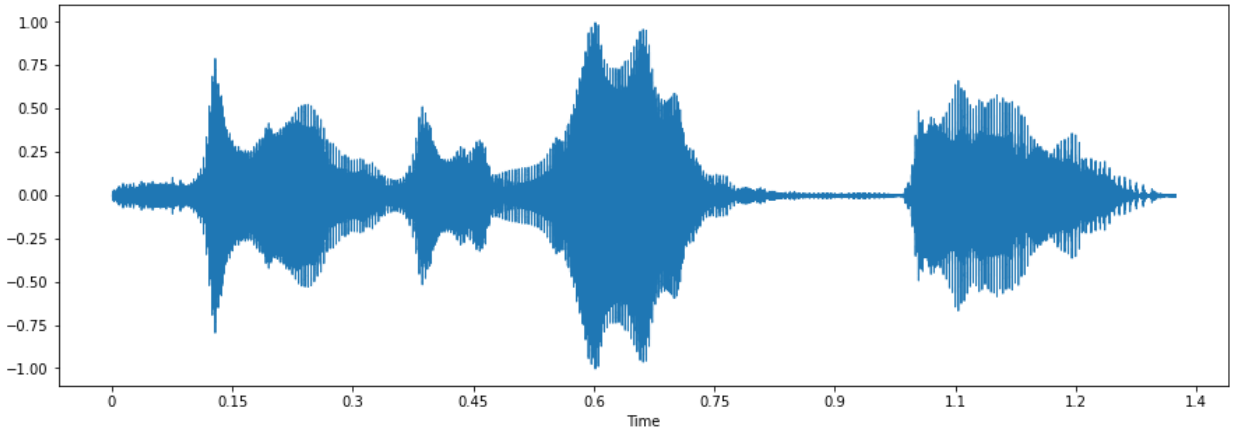

Fig. III.1.31. - Example of signal of a sample with the “anger” label

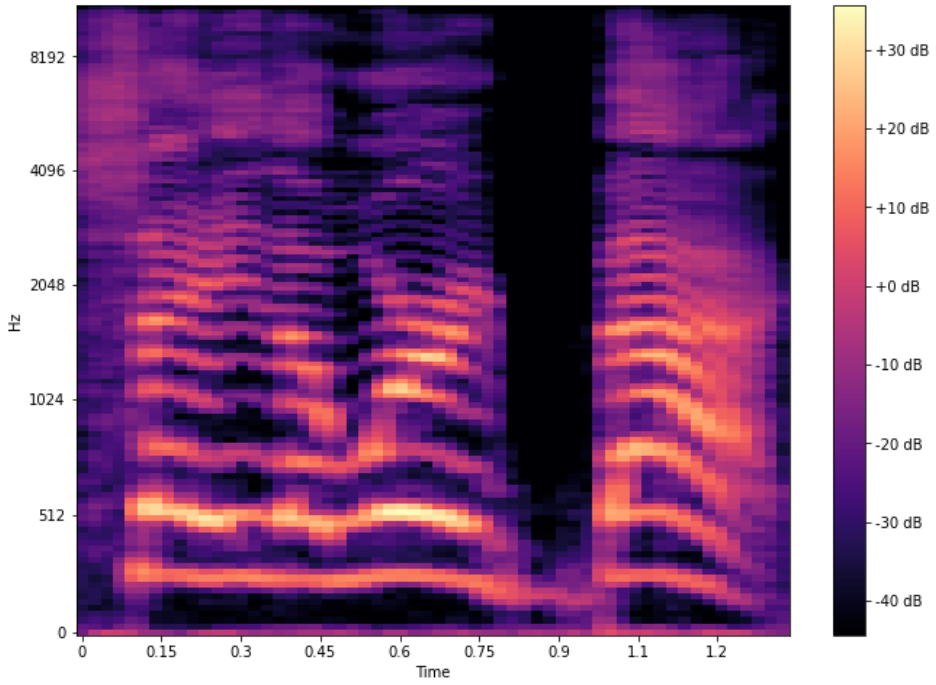

Fig. III.1.32. - Example of Mel Spectrogram of a sample with the “anger” label

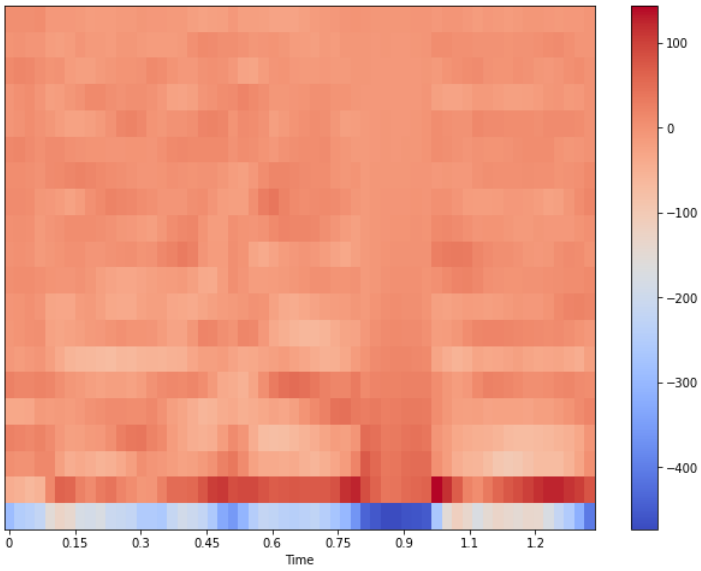

Fig. III.1.33. - Example of MFCC of a sample with the “anger” label

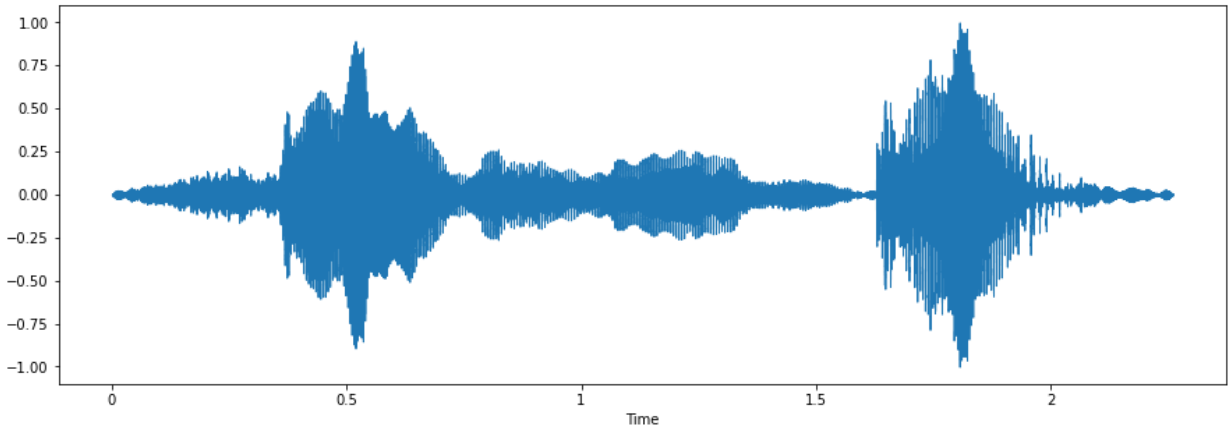

Fig. III.1.34. - Example of signal of a sample with the “disgust” label

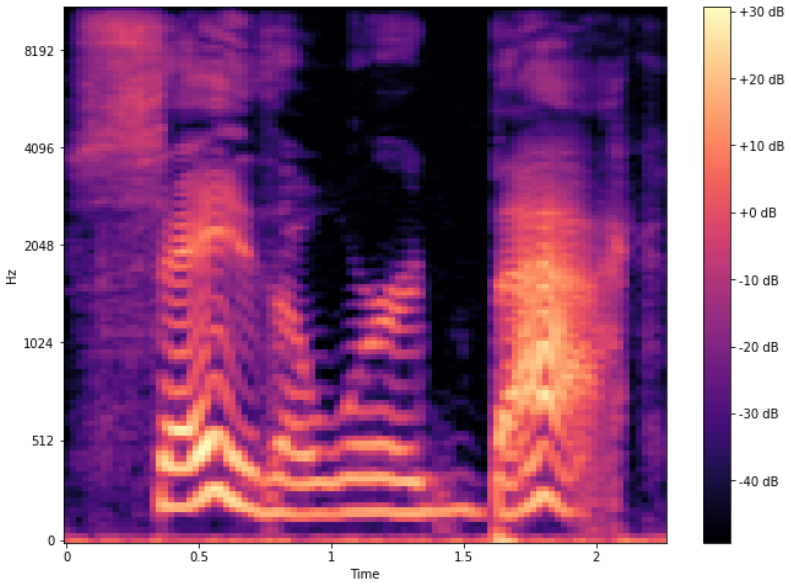

Fig. III.1.35. - Example of Mel Spectrogram of a sample with the “disgust” label

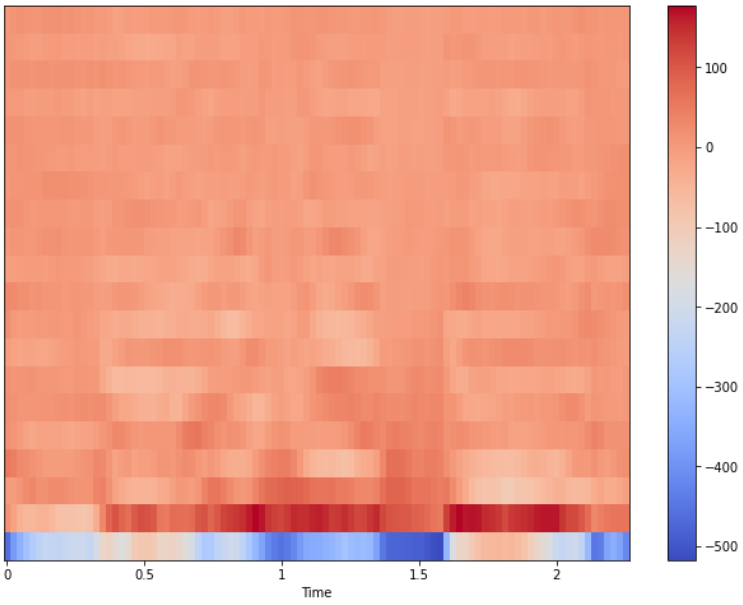

Fig. III.1.36. - Example of MFCC of a sample with the “disgust” label

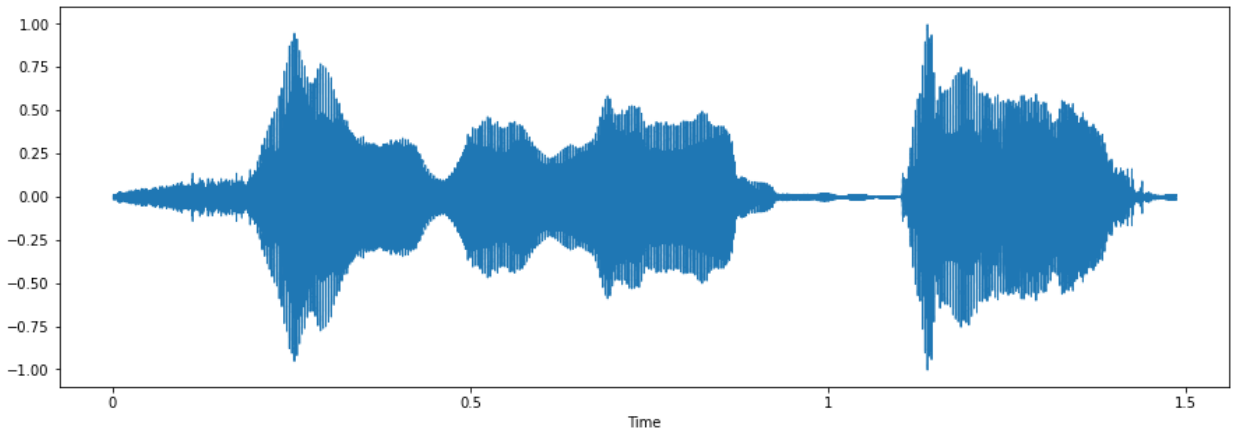

Fig. III.1.37. - Example of signal of a sample with the “fear” label

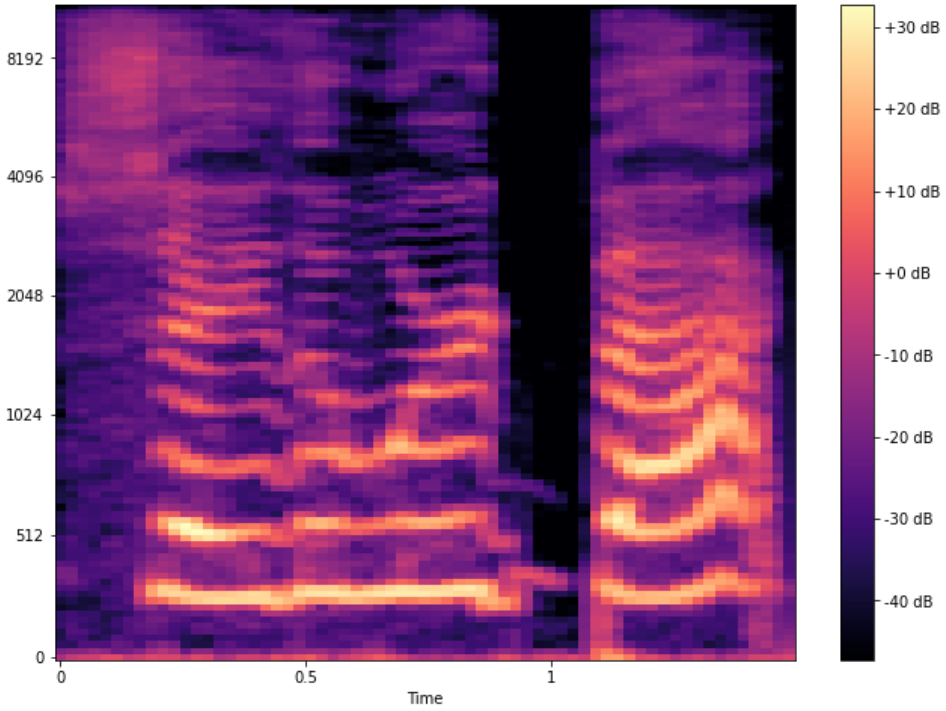

Fig. III.1.38. - Example of Mel Spectrogram of a sample with the “fear” label

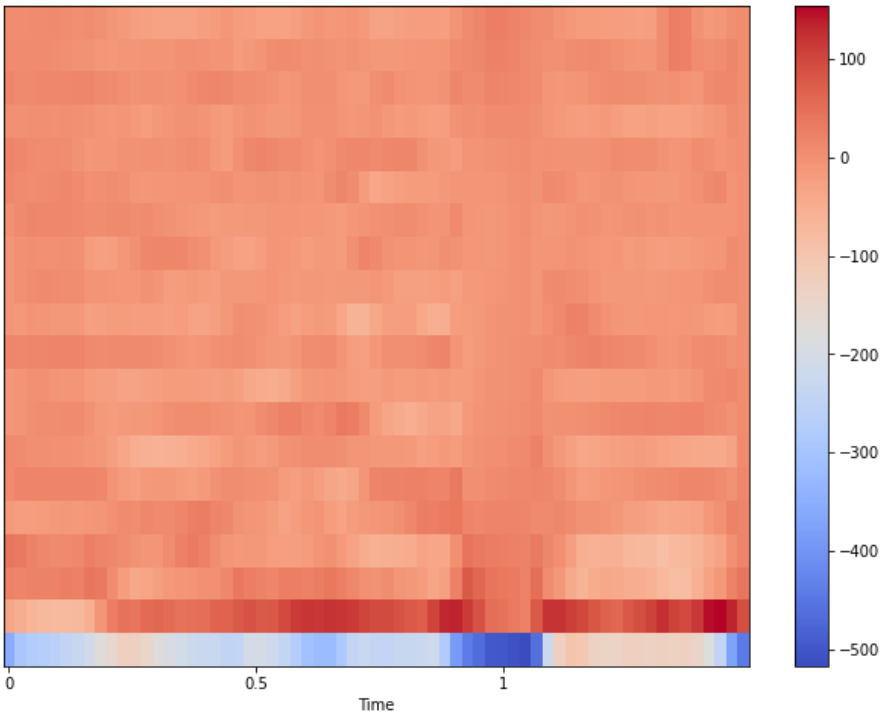

Fig. III.1.39. - Example of MFCC of a sample with the “fear” label

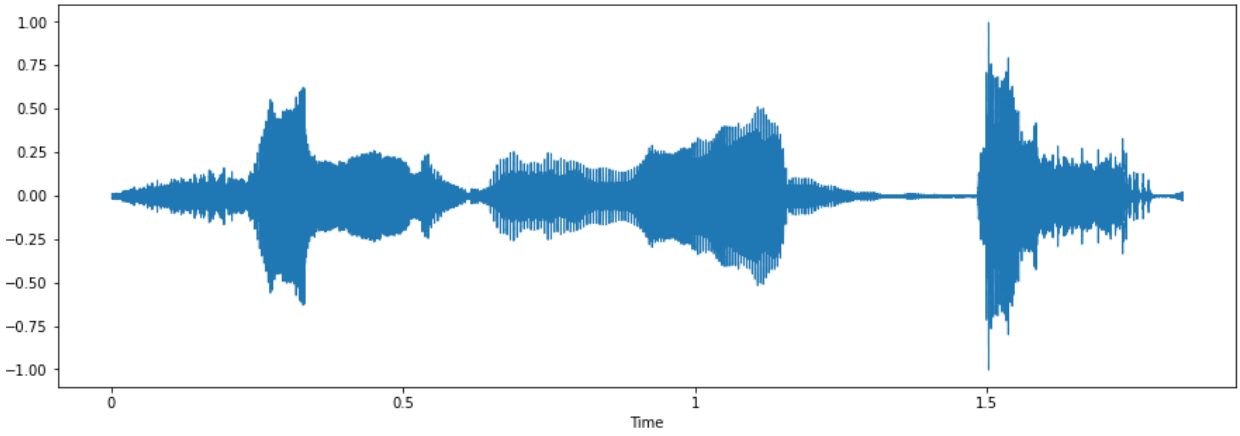

Fig. III.1.40. - Example of signal of a sample with the “happiness” label

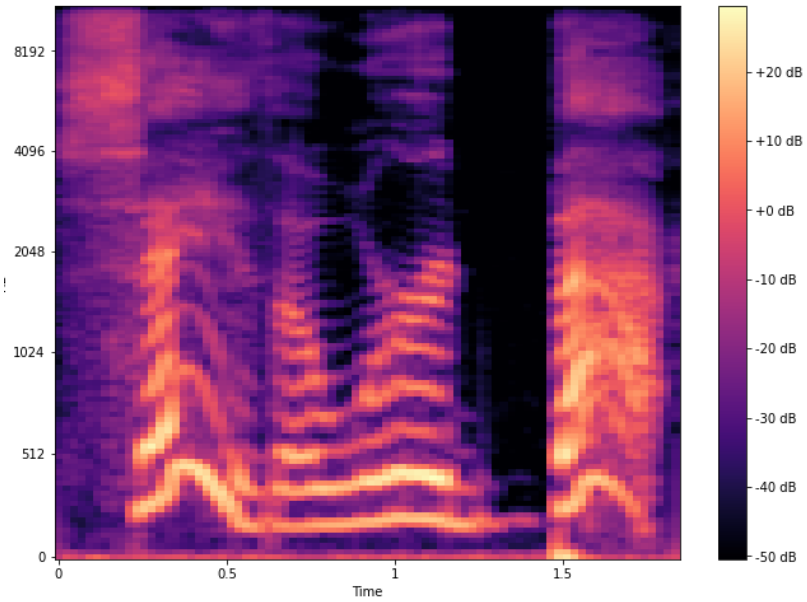

Fig. III.1.41. - Example of Mel Spectrogram of a sample with the “happiness” label

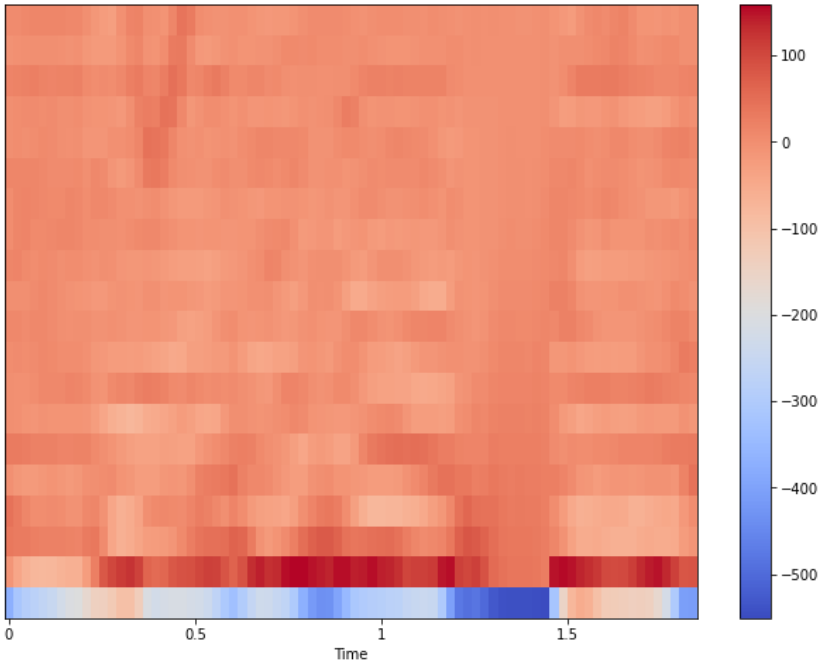

Fig. III.1.42. - Example of MFCC of a sample with the “happiness” label

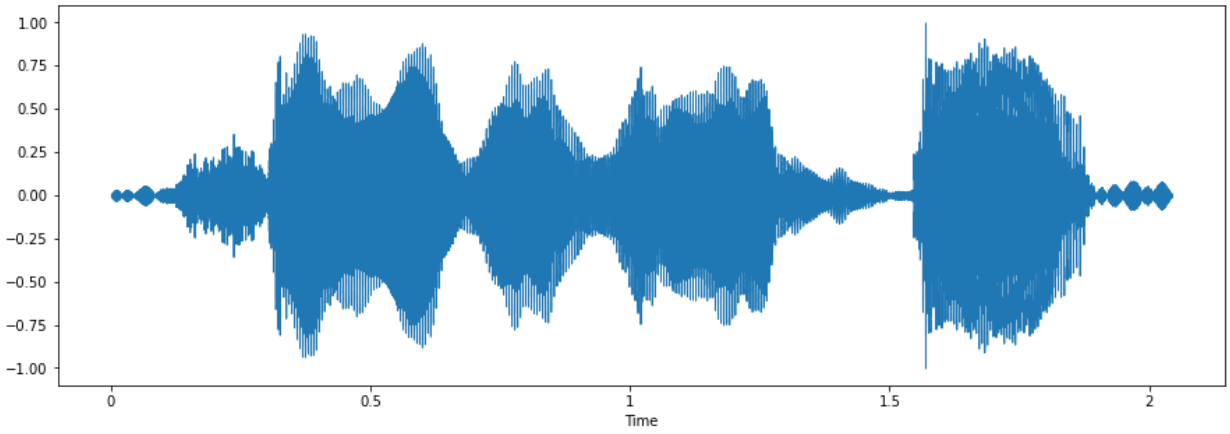

Fig. III.1.43. - Example of signal of a sample with the “neutral” label

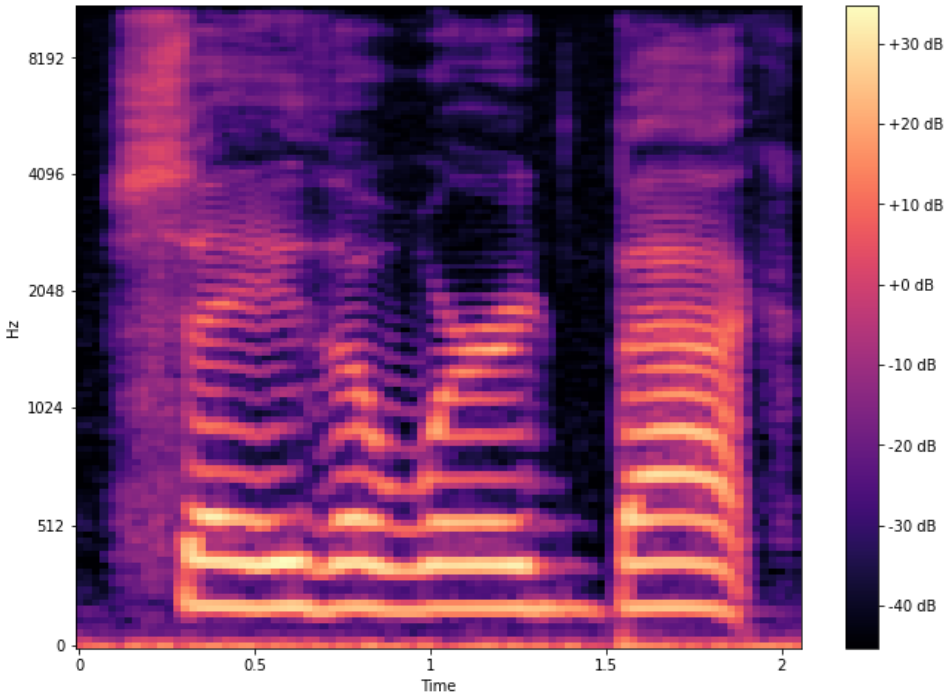

Fig. III.1.44. - Example of Mel Spectrogram of a sample with the “neutral” label

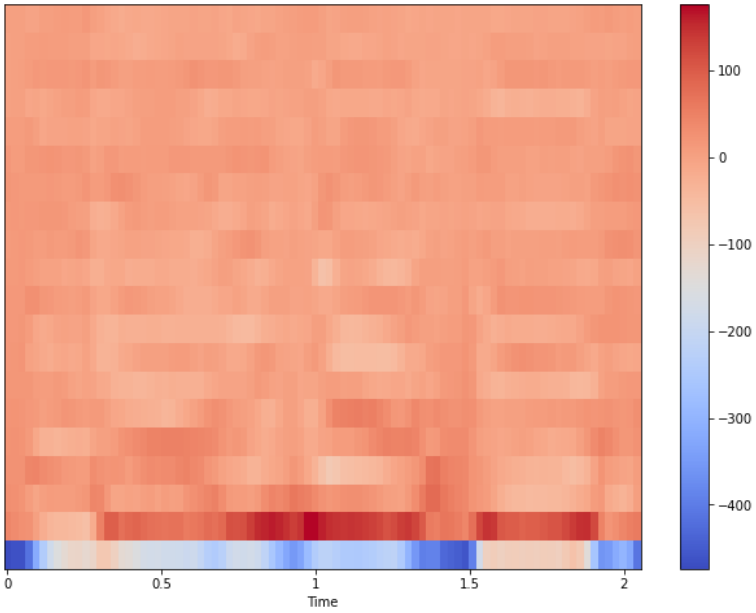

Fig. III.1.45. - Example of MFCC of a sample with the “neutral” label

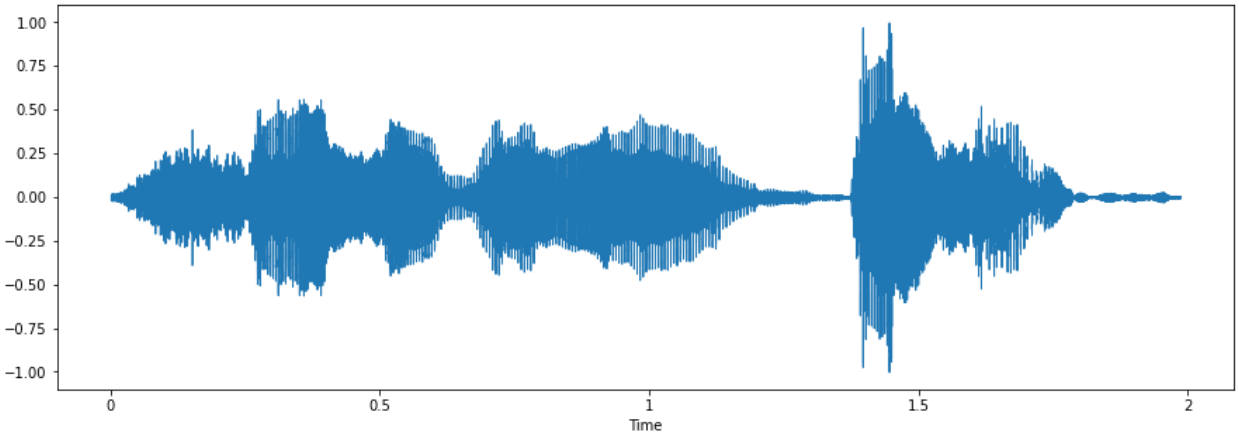

Fig. III.1.46. - Example of signal of a sample with the “surprise” label

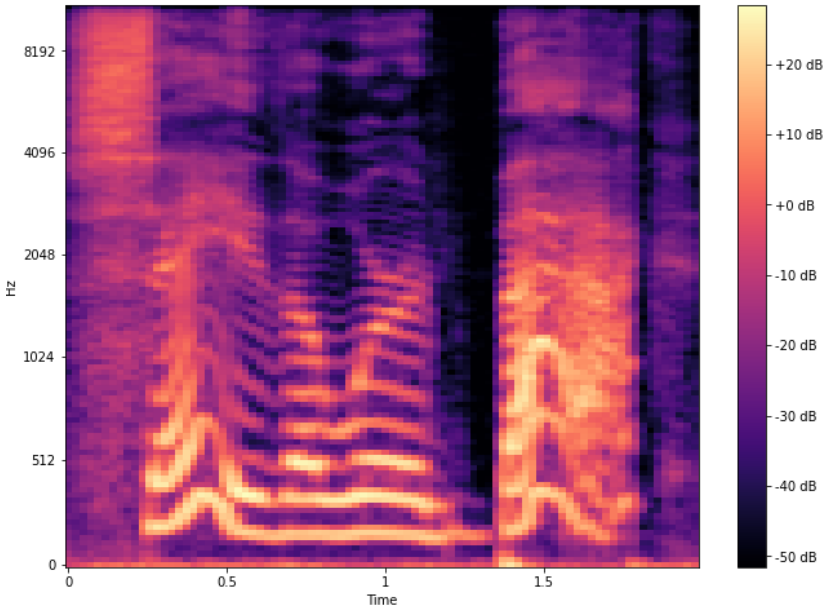

Fig. III.1.47. - Example of Mel Spectrogram of a sample with the “surprise” label

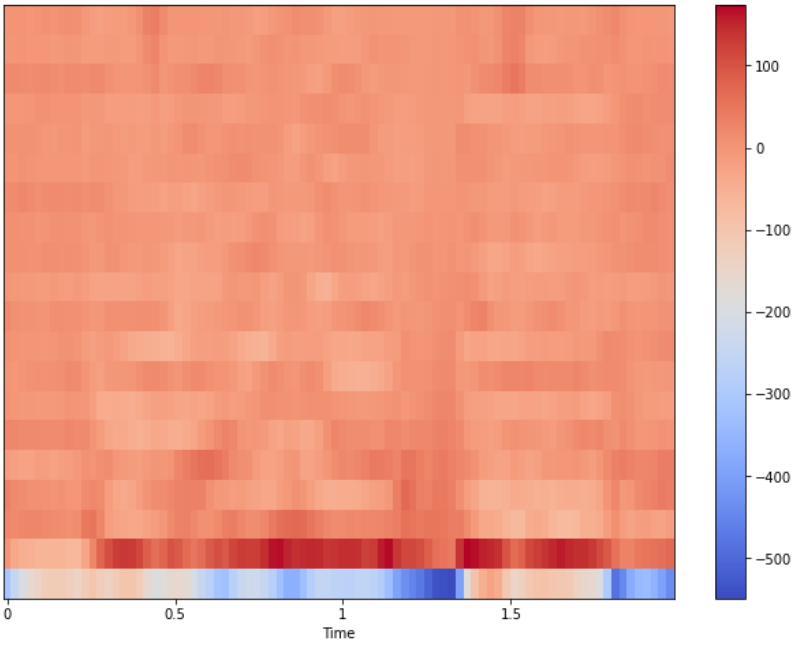

Fig. III.1.48. - Example of MFCC of a sample with the “surprise” label

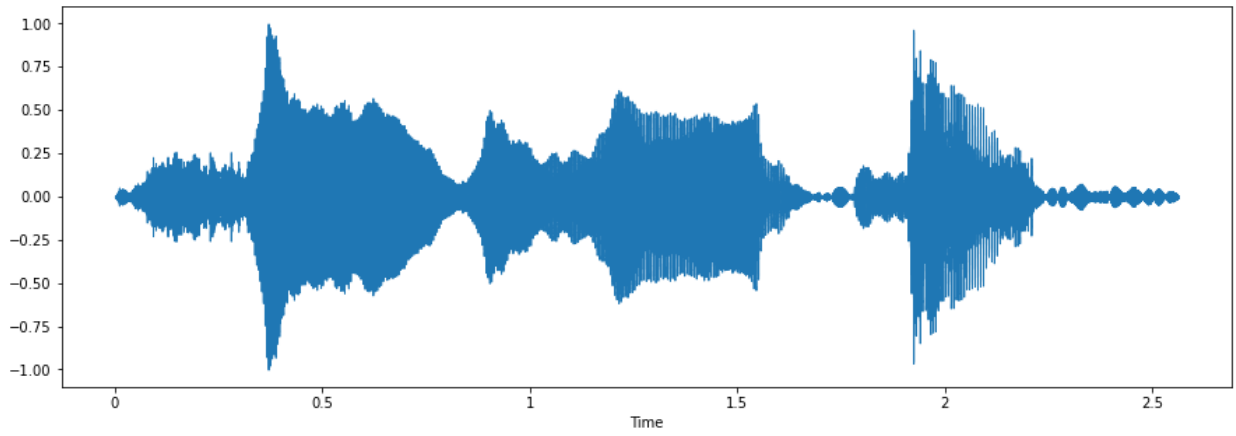

Fig. III.1.49. - Example of signal of a sample with the “sadness” label

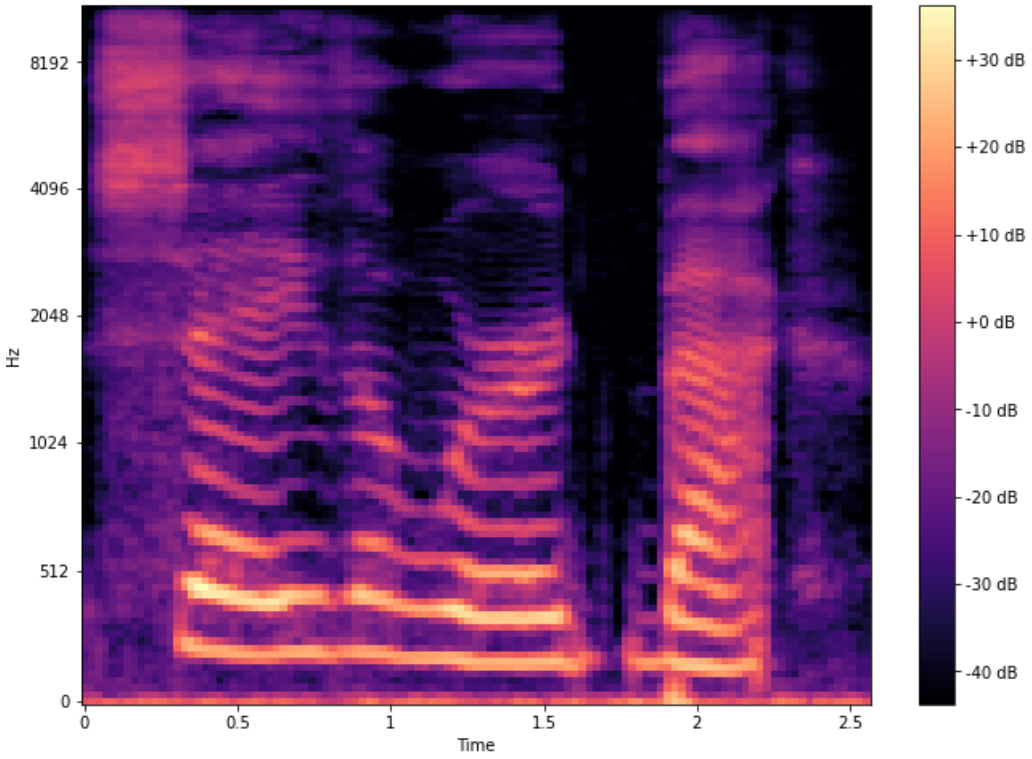

Fig. III.1.50. - Example of Mel Spectrogram of a sample with the “sadness” label

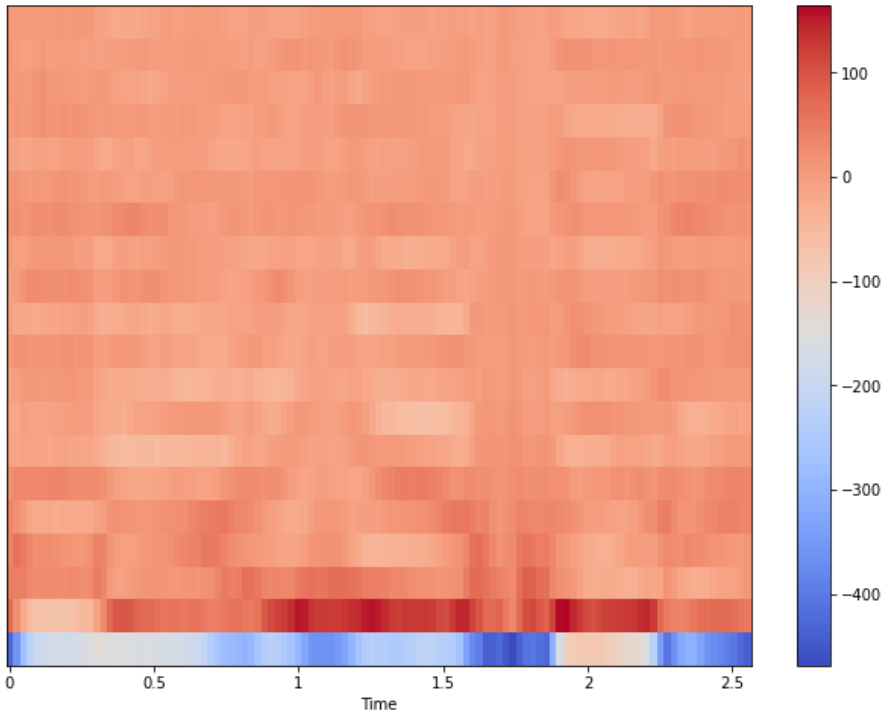

Fig. III.1.51. - Example of MFCC of a sample with the “sadness” label

3.2. Preprocessing

The labels and the paths of the files were saved into a Pandas Dataframe. Using the paths column, the signals for each file were read using librosa. The signal rate of all the files was not the same throughout the dataset so when loading them with librosa they were also resampled to 22050 Hz as most samples had this signal rate.

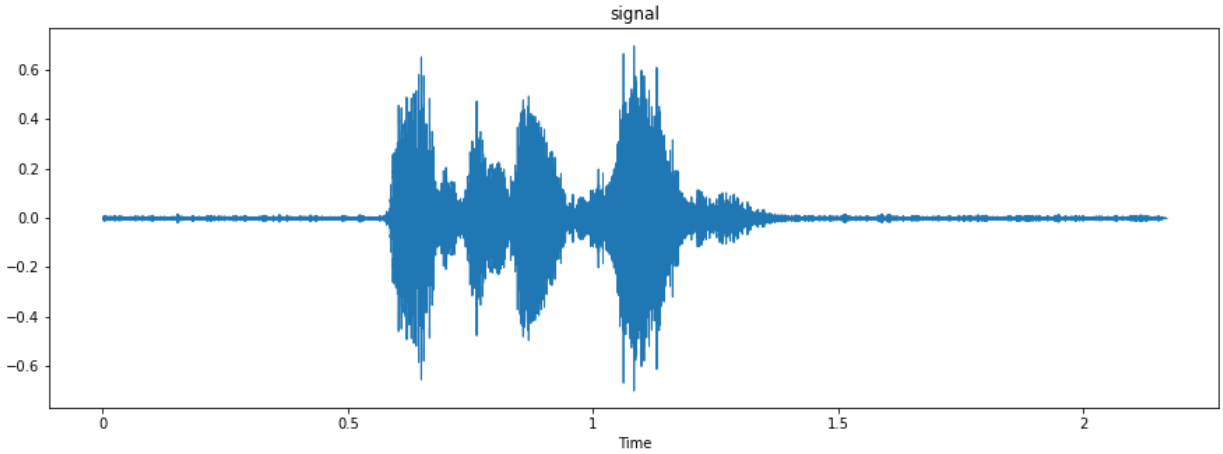

Fig. III.2.1. Example of signal read by librosa

After visualizing some samples, normalization was necessary. Even though librosa normalizes the samples when they are loaded, resampling happens afterwards and that changes the data significantly so librosa normalization is used on the data again.

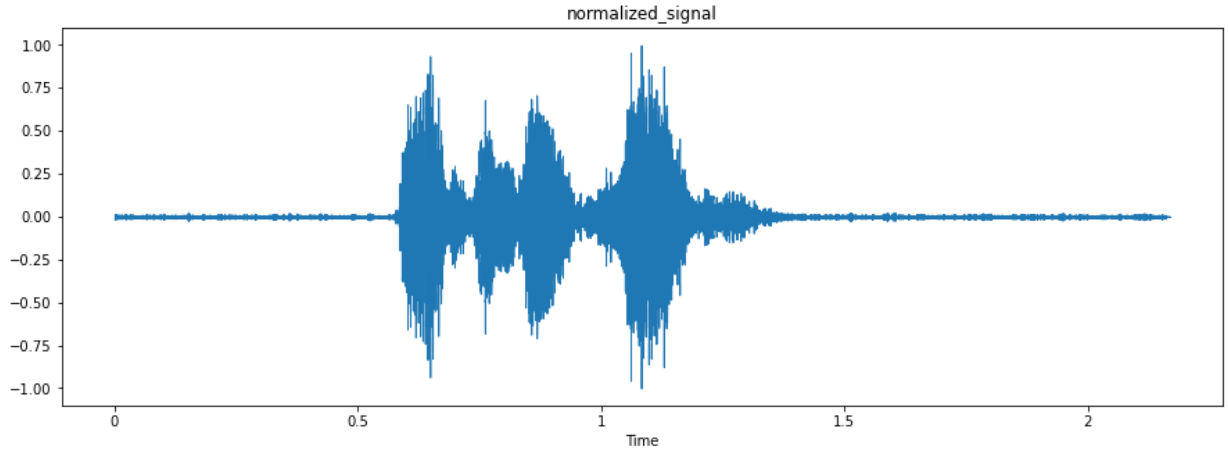

Fig. III.2.2. Example of normalized signal

Because all samples had a portion of silence at the start and at the end, all samples were trimmed using librosa in order to get rid of useless data and shorten the lengths of the signals.

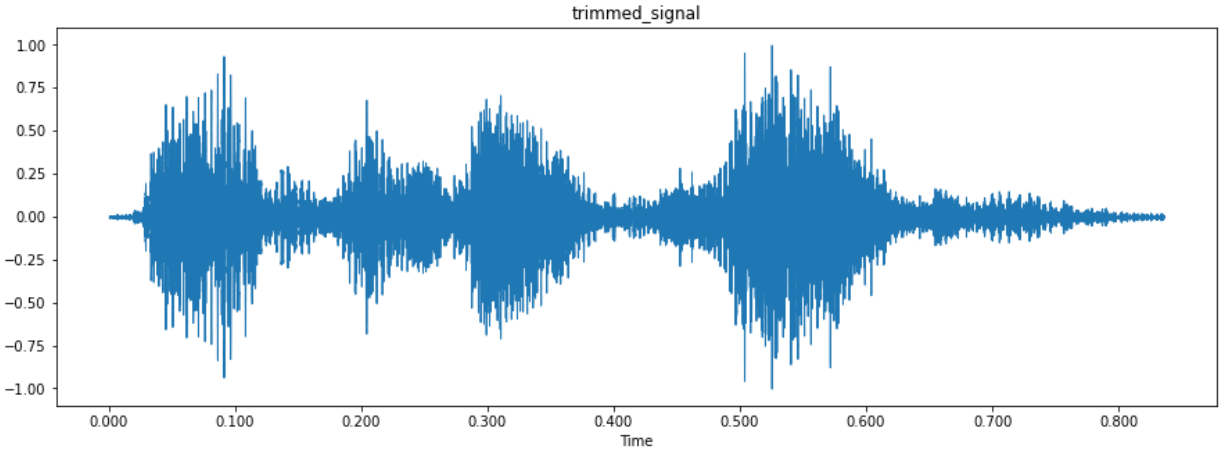

Fig. III.2.3. Example of trimmed signal

At this point the data was normalized and trimmed but the fact that all samples had different lengths made it impossible to load them into a model, so the max length of all the data was computed, saved and used to pad all signals to this length. In some cases in the data augmentation process the padding was added after the augmentation in order for the augmented data to make sense.

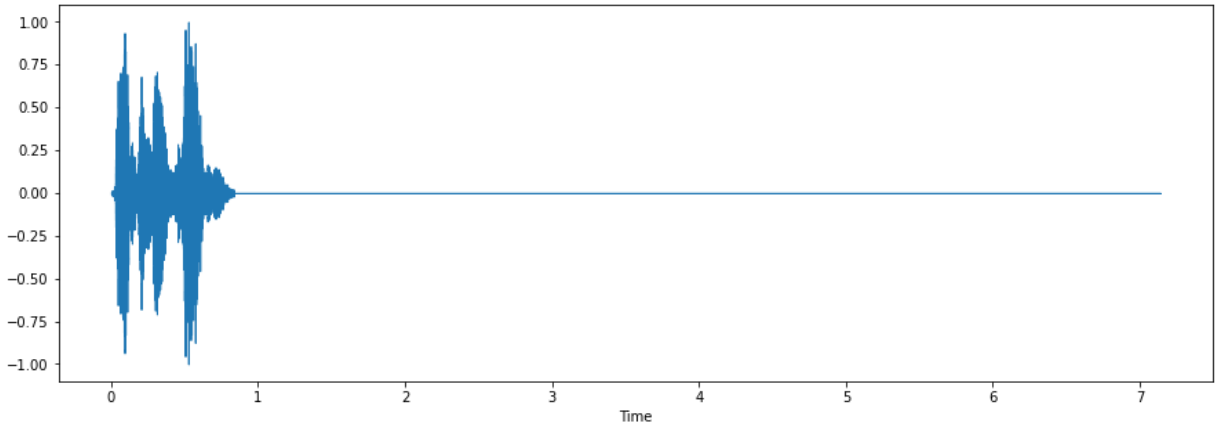

Fig. III.2.4. Example of padded signal

3.3. Data Augmentation

For data augmentation, a range of methods were applied on the data before feature extraction and a few methods were applied after the feature extraction. Data augmentation was applied after the train, validation, test split and only on the train data in order to avoid cross contamination between the sets, making sure that no version of a sample in the train set ends up in either the validation set or the test set.

3.3.1. Before feature extraction

a) Noise addition - this method adds noise to the signal creating a different sample of the original signal. Noise addition was performed on the unpadded data as adding noise to the pad would be useless and possibly confuse the model. The padding was added after the noise was added.
b) Pitch shifting - this method changes the octave of the sound making the signal thicker or thinner. For this process an effect from librosa was used.
c) Time shifting - this method cuts a segment of the signal and moves it somewhere else on the time axis. In order for the resulted signal to make sense, padding was added before this process
d) Time stretching - this method simply makes the sound faster or slower. Both the sped up and the slowed down versions of the original signal were used.

3.3.2 After feature extraction

a) Time mask - this method hides a number of channels on the time axis
b) Frequency mask - this method hides a number of channels on the frequency axis

3.4. Feature Extraction

The signal can be the input for a model as it is, but in order to compare and possibly improve results different feature extraction methods and combinations of methods were used on the signal.

a) Mel Spectrogram - for each sample the Mel Spectrogram was extracted using librosa. The mel spectrogram is a visualization of the frequency of a signal, converted to the Mel scale. The Mel scale is a unit of pitch that is better for the human ear to distinguish between frequencies.
b) MFCC - The MFCC is a method of extracting features that give information about the shape of a spectral envelope.
c) Deltas - The Delta and Delta-delta are the differences of signal features
d) Root Mean Square - The RMS is the total magnitude of the signal which represents the energy of the sound
e) ZCR - The Zero Crossing Rate is to the rate at which a signal moves from positive to negative through zero and the other way around

For this project the Mel Spectrogram and the MFCC were used separately as input for different models both as they are and in the form of an image. The MFCC was also used in combination with the Deltas for one experiment and in combination with both RMS and ZCR for another experiment. All of the variations of the feature extraction method combinations were tested on both unaugmented and augmented data. In the cases where the Mel Spectrogram or the MFCC were treated as images, a third dimension needed to be added, thus changing the shape in order to be the input for a Conv2D layer.

IV. Models

For the purpose of this project 4 different model architectures were selected for comparison. The first 3 are all models that take two dimensional data as input and these were used for all the combinations of data resulting from the original data after feature extraction and in some cases after augmentation. The axes of the data were swapped in order to follow the time axis. For all models the optimizer used is “adam”.

Model optimization was done with the use of the early stopping technique, implemented with Keras’s callbacks. This technique is used to stop the neural network from iterating through the epochs once the chosen metric, in this case, validation loss, does not improve. The technique then restores the best weights encountered during any epoch, according to the metric. This procedure reduces training time and improves performance.

Another optimization technique used was the reduction of the learning rate on plateau. This consists of lowering the learning rate once the improvement in the chosen metric stops improving significantly (or “plateaus”). This helps the model generalize better, by adjusting the model weights less with each new sample, keeping a baseline and preventing overfitting. The metric tracked here was, again, validation loss.

The emotions were encoded as labels using OneHotEncoder. This outputs an array of length equal to the number of different values in the labels. All of the values of this array are zeros with the exception of one position which is one and represents the emotion encoded as a class. The loss function used for all the models is “categorical_crossentropy”.

4.1. CNN + LSTM

The first model combines the layers used in a convolutional neural network and a LSTM. It contains 2 Conv2D layers followed by BatchNormalization and Dropout, 2 LSTM layers followed by BatchNormalization and Dropout and 2 Dense layers before the final Dense layer. The activation function for all convolutional and dense layers is “relu”, the only exception being the last dense layer where the activation function is “softmax". The architecture is shown in Fig. IV.1.1.

4.2. LSTM

This model is a basic LSTM model with only 2 LSTM layers and a Dense output layer. The architecture can be observed in Fig. IV.2.1.

4.3. CNN - Conv1D

The third model contains 4 Conv1D layers with BatchNormalization, AveragePooling1D and Dropout in between, followed by 1 Dense layer, a Dropout and a BatchNormalization before the final Dense output layer. “Relu” is the activation function for all of the convolutional and dense layers except for the last dense layer where the activation function is “softmax”. The architecture can be observed in Fig. IV.3.1.

4.4. CNN - Conv2D

The final takes 3 dimensional data as input. It is used for the Mel Spectrograms and the MFCCs when they are treated as images and have a third dimension added. It has the same architecture as the Conv1D model but all the Conv1D layers are replaced with Conv2D layers and all the AveragePooling1D layers are replaced with AveragePooling2D layers. The architecture can be observed in Fig. IV.4.1.

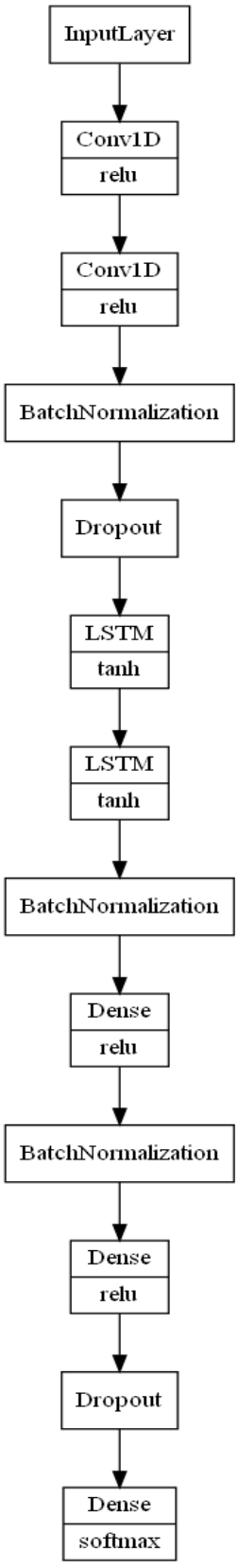

Fig. IV.1.1

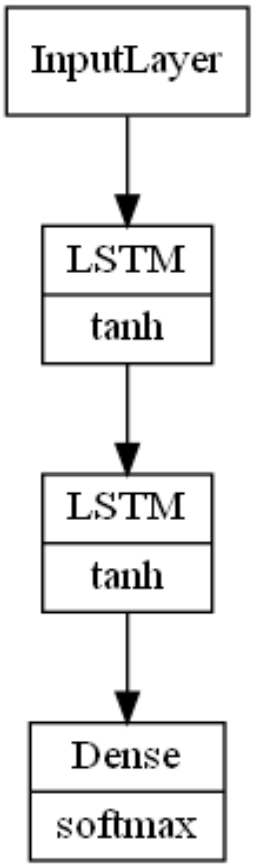

Fig. IV.2.1

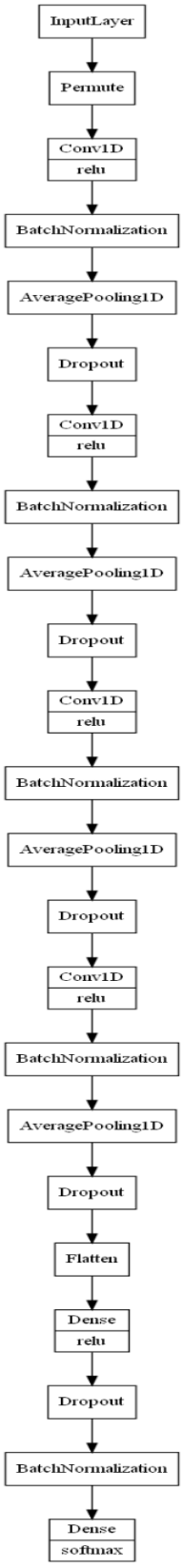

Fig. IV.3.1

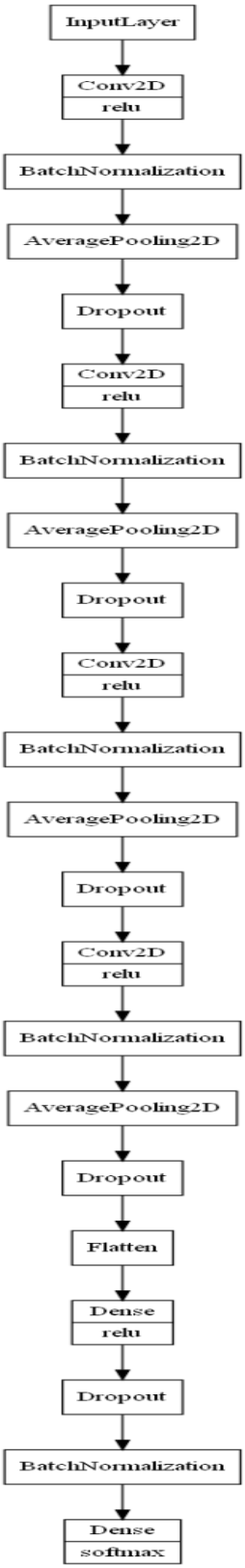

Fig. IV.4.1

V. Results

5.1. Set 1 - all samples, 7 labels

Set 1 - all samples 7 accuracy	CNN+LSTM		LSTM		Conv1D		Conv2D	
	val	test	val	test	val	test	val	test
MFCC	58	58	50	52	67	68	-	-
Aug. MFCC	62	62	55	55	71	71	-	-
Mask MFCC	59	60	46	50	68	67	-	-
Aug. + Mask MFCC	64	60	56	54	69	68	-	-
MFCC+Delta	54	53	42	47	65	67	-	-
Aug. MFCC+Delta	64	57	53	52	71	71	-	-
Mask MFCC+Delta	50	49	44	44	70	69	-	-
Aug.+Mask MFCC+Delta	58	58	49	48	71	70	-	-
MFCC+RMS+ZCR	59	54	52	52	68	66	-	-
Aug. MFCC+RMS+ZCR	64	62	56	54	67	68	-	-
Log Mel Spec	52	52	50	50	69	66	-	-
Aug. Log Mel Spec	56	56	53	53	76	73	-	-
Mask Log Mel Spec	58	52	51	48	70	68	-	-
Aug.+Mask Log Mel Spec	50	54	46	52	67	72	-	-
Image MFCC	-	-	-	-	-	-	72	71
Image Aug. MFCC	-	-	-	-	-	-	73	71
Image Log Mel Spec	-	-	-	-	-	-	72	72
Image Aug. Log Mel Spec	-	-	-	-	-	-	71	70

Table 5.5.1.1.

5.2. Set 2 - male samples

Set 2 - male accuracy	CNN+LSTM		LSTM		Conv1D		Conv2D	
	val	test	val	test	val	test	val	test
MFCC	38	43	35	38	52	55	-	-
Aug. MFCC	48	45	46	39	58	56	-	-
Mask MFCC	40	33	33	34	58	57	-	-
Aug. + Mask MFCC	47	46	37	43	60	55	-	-
MFCC+Delta	28	30	29	33	51	57	-	-
Aug. MFCC+Delta	43	38	30	37	56	60	-	-
Mask MFCC+Delta	32	26	26	31	56	56	-	-
Aug.+Mask MFCC+Delta	38	42	29	32	61	61	-	-
MFCC+RMS+ZCR	36	42	32	37	55	58	-	-
Aug. MFCC+RMS+ZCR	49	49	39	41	60	59	-	-
Log Mel Spec	36	34	32	32	55	52	-	-
Aug. Log Mel Spec	37	38	37	39	63	61	-	-
Mask Log Mel Spec	33	36	34	36	51	49	-	-
Aug.+Mask Log Mel Spec	37	30	39	39	58	61	-	-
Image MFCC	-	-	-	-	-	-	57	56
Image Aug. MFCC	-	-	-	-	-	-	64	65
Image Log Mel Spec	-	-	-	-	-	-	56	53
Image Aug. Log Mel Spec	-	-	-	-	-	-	60	61

Table 5.5.2.2

5.3. Set 3 - female samples

Set 3 - female accuracy	CNN+LSTM		LSTM		Conv1D		Conv2D	
	val	test	val	test	val	test	val	test
MFCC	65	66	57	57	77	75	-	-
Aug. MFCC	71	72	60	65	77	77	-	-
Mask MFCC	67	67	60	58	77	76	-	-
Aug. + Mask MFCC	70	64	63	63	79	77	-	-
MFCC+Delta	55	52	54	52	74	75	-	-
Aug. MFCC+Delta	70	67	59	55	78	78	-	-
Mask MFCC+Delta	60	54	49	50	74	77	-	-
Aug.+Mask MFCC+Delta	71	64	57	57	77	78	-	-
MFCC+RMS+ZCR	66	67	59	60	78	78	-	-
Aug. MFCC+RMS+ZCR	72	69	65	61	77	78	-	-
Log Mel Spec	60	58	56	52	77	74	-	-
Aug. Log Mel Spec	62	63	61	58	75	79	-	-
Mask Log Mel Spec	62	57	59	53	79	72	-	-
Aug.+Mask Log Mel Spec	66	64	62	58	81	81	-	-
Image MFCC	-	-	-	-	-	-	79	79
Image Aug. MFCC	-	-	-	-	-	-	80	77
Image Log Mel Spec	-	-	-	-	-	-	77	76
Image Aug. Log Mel Spec	-	-	-	-	-	-	79	79

Table 5.5.3.3.

5.4. Set 4 - all samples, 14 labels

Set 4 - all samples 14 accuracy	CNN+LSTM		LSTM		Conv1D		Conv2D	
	val	test	val	test	val	test	val	test
MFCC	53	51	49	51	64	67	-	-
Aug. MFCC	60	59	48	50	68	70	-	-
Mask MFCC	55	53	42	43	67	67	-	-
Aug. + Mask MFCC	60	56	51	50	69	67	-	-
MFCC+Delta	47	44	35	34	66	67	-	-
Aug. MFCC+Delta	59	61	44	42	68	71	-	-
Mask MFCC+Delta	48	42	34	38	65	68	-	-
Aug.+Mask MFCC+Delta	59	59	42	40	69	70	-	-
MFCC+RMS+ZCR	55	51	41	42	67	67	-	-
Aug. MFCC+RMS+ZCR	60	59	50	48	65	67	-	-
Log Mel Spec	44	41	40	37	69	66	-	-
Aug. Log Mel Spec	58	55	47	45	70	70	-	-
Mask Log Mel Spec	44	40	38	36	71	68	-	-
Aug.+Mask Log Mel Spec	59	58	50	49	68	73	-	-
Image MFCC	-	-	-	-	-	-	70	68
Image Aug. MFCC	-	-	-	-	-	-	71	74
Image Log Mel Spec	-	-	-	-	-	-	70	72
Image Aug. Log Mel Spec	-	-	-	-	-	-	73	72

Table 5.5.4.4.

5.5 Metrics

For the purpose of comparing more in-depth metrics, the top performing feature set and model were selected for each set and for each data structure: mfcc, mel-spectrogram or image. The metrics used were: loss, accuracy of the model, confusion matrix, and accuracy, precision, recall, sensitivity, specificity and f1 score per class, in a one versus all style. Since this is a classification model outside of a binary classification, these metrics are useful for determining the performance because class imbalance is present.

The loss and accuracy of the model are the parameters calculated during training and validation.

The confusion matrix is created by using the model to predict the test set. The true positives (tp), false positives (fp), true negatives (tn) and false negatives (fn) are extracted for each class and with these we can calculate the following metrics:

$$Accuracy = (tp + tn) / (tp + fp + fn + tn)$$

$$Precision = tp / (tp + fp)$$

$$Recall = tp / (tp + fn)$$

$$Sensitivity = tp / (tp + fn)$$

$$Specificity = fp / (fp + tn)$$

$$F1\ Score = 2 * (precision * recall) / (precision + recall)$$

Set 1 - all samples, 7 labels

Model: Conv1D

Features: Augmented MFCC + Delta

Validation accuracy: 71

Test accuracy: 71

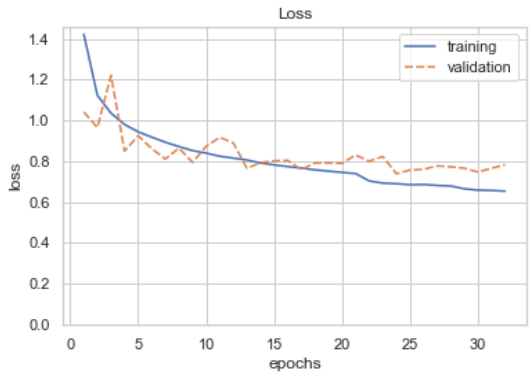

Fig. V.5.1

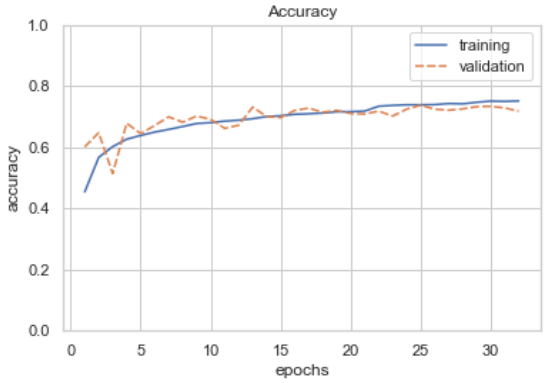

Fig. V.5.2

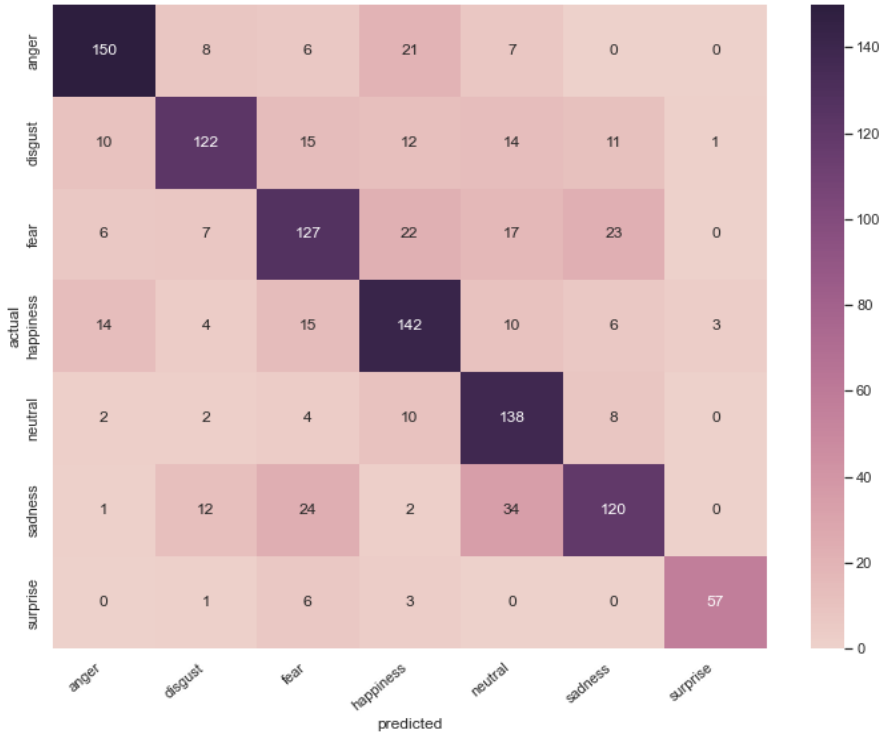

Fig. V.5.3

	class	accuracy	precision	recall	f1_score	sensitivity	specificity
0	anger	0.937343	0.819672	0.781250	0.800000	0.781250	0.032836
1	disgust	0.918964	0.782051	0.659459	0.715543	0.659459	0.033597
2	fear	0.878864	0.644670	0.628713	0.636591	0.628713	0.070352
3	happiness	0.898079	0.669811	0.731959	0.699507	0.731959	0.069791
4	neutral	0.909774	0.627273	0.841463	0.718750	0.841463	0.079380
5	sadness	0.898914	0.714286	0.621762	0.664820	0.621762	0.047809
6	surprise	0.988304	0.934426	0.850746	0.890625	0.850746	0.003540

Fig. V.5.4

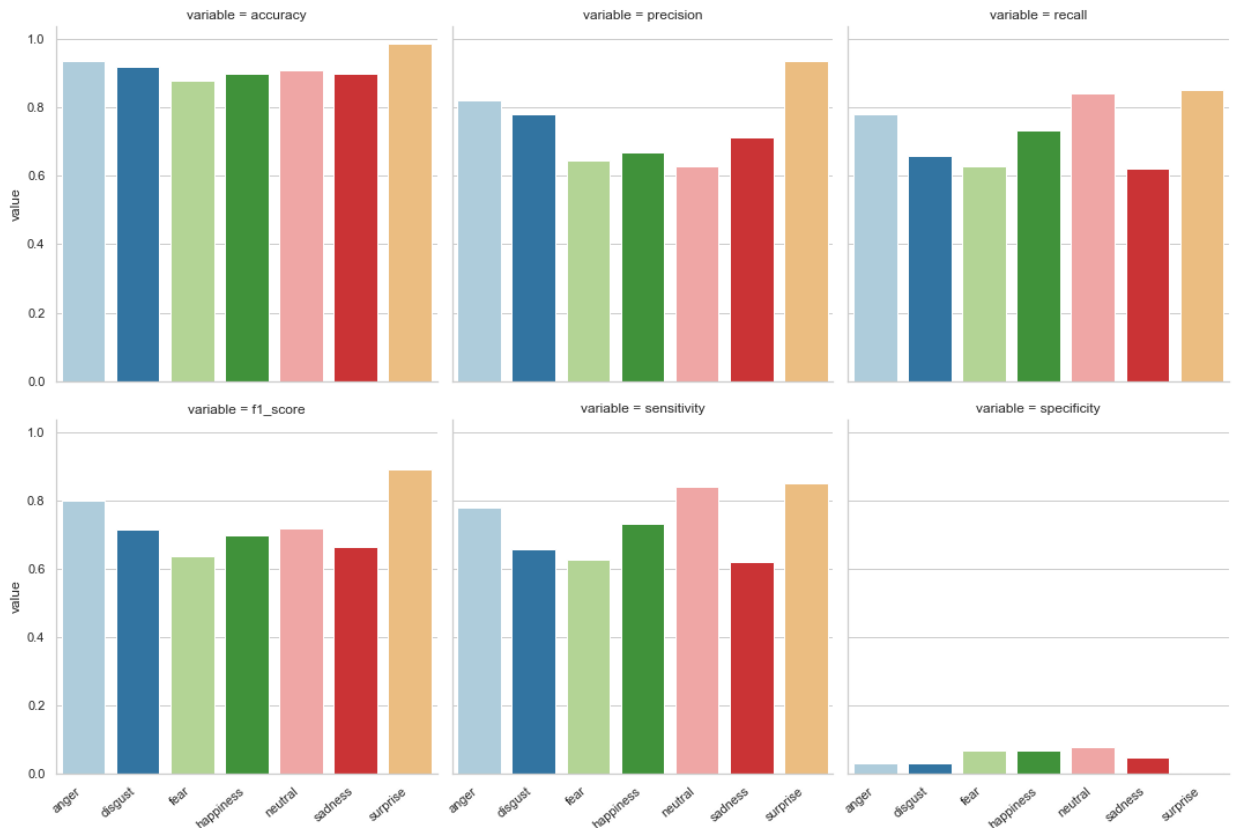

Fig. V.5.5

Model: Conv1D

Features: Augmented Log Mel-Spectrogram

Validation accuracy: 76

Test accuracy: 73

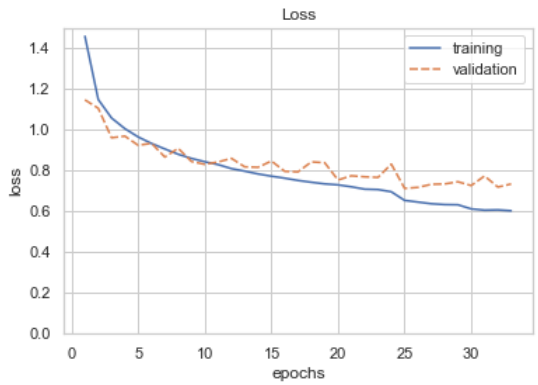

Fig. V.5.6

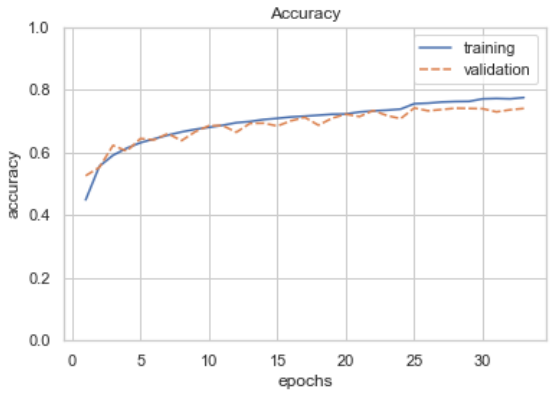

Fig. V.5.7

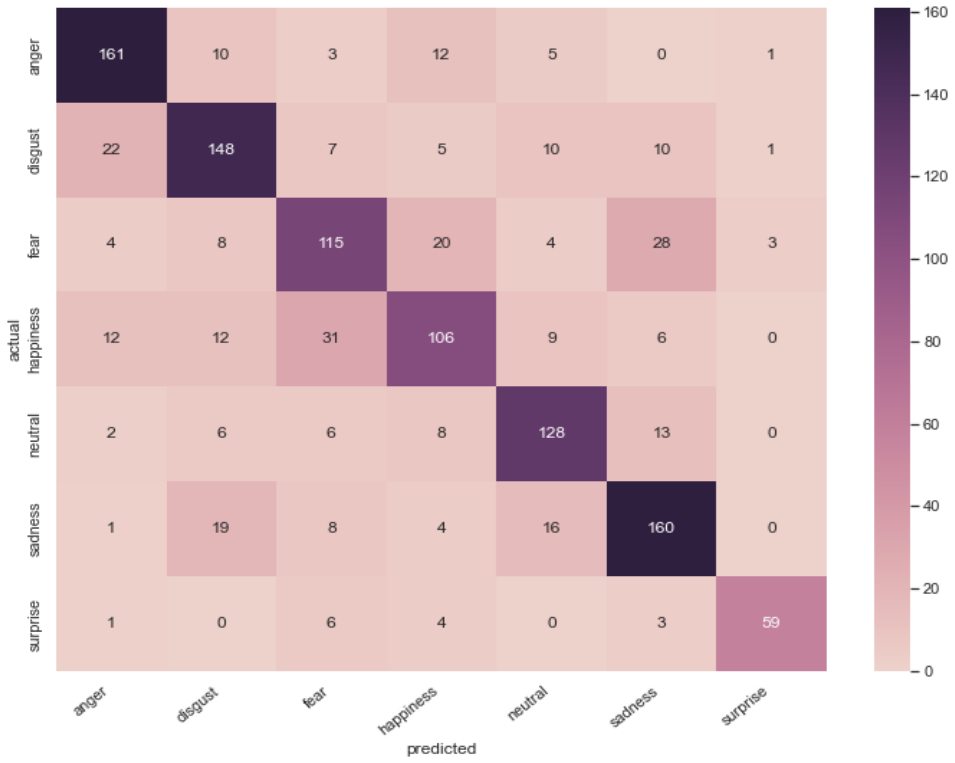

Fig. V.5.8

	class	accuracy	precision	recall	f1_score	sensitivity	specificity
0	anger	0.939014	0.793103	0.838542	0.815190	0.838542	0.041791
1	disgust	0.908104	0.729064	0.729064	0.729064	0.729064	0.055332
2	fear	0.893066	0.653409	0.631868	0.642458	0.631868	0.060099
3	happiness	0.897243	0.666667	0.602273	0.632836	0.602273	0.051910
4	neutral	0.934002	0.744186	0.785276	0.764179	0.785276	0.042553
5	sadness	0.909774	0.727273	0.769231	0.747664	0.769231	0.060667
6	surprise	0.984127	0.921875	0.808219	0.861314	0.808219	0.004448

Fig. V.5.9

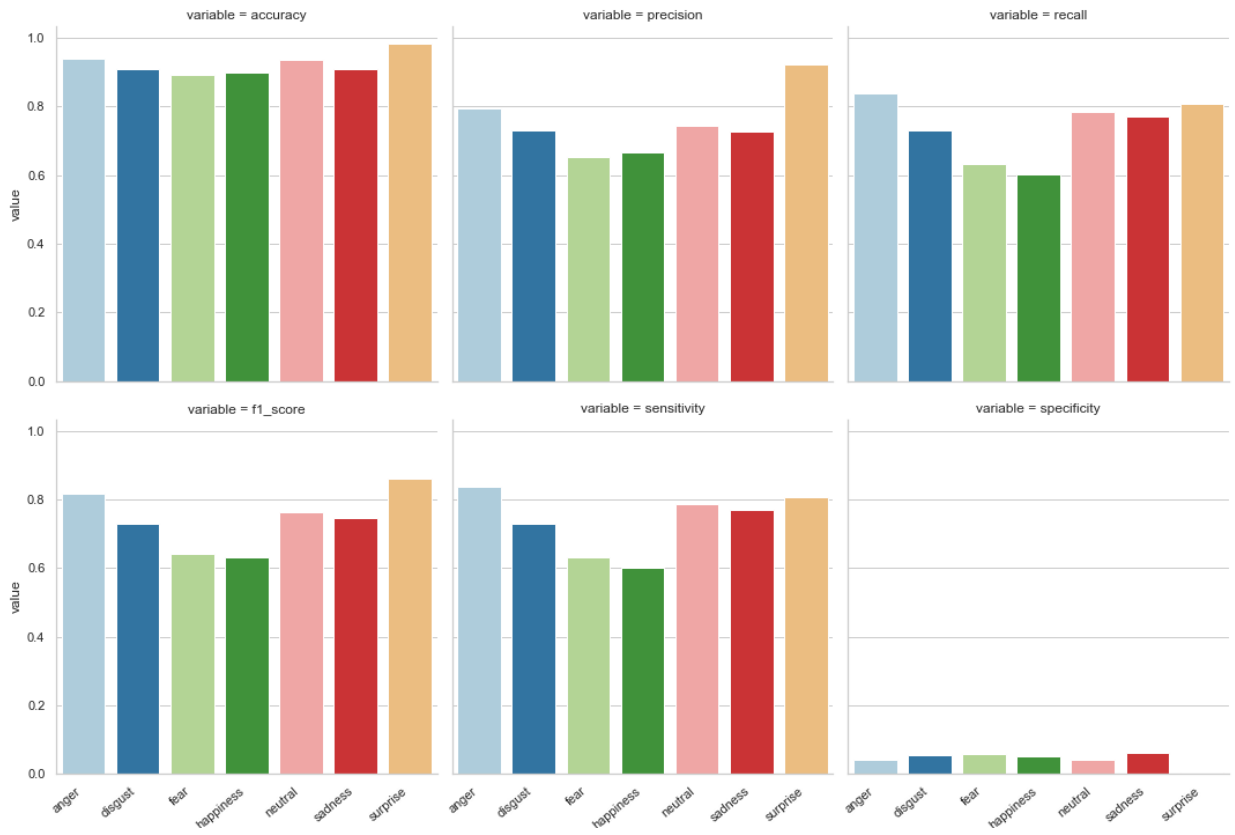

Fig. V.5.10

Model: Conv2D

Features: Image Log Mel-Spectrogram

Validation accuracy: 72

Test accuracy: 72

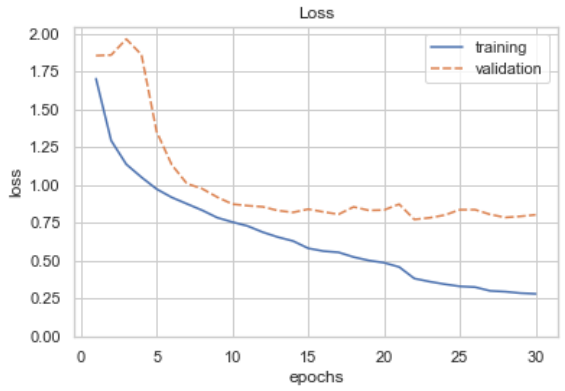

Fig. V.5.11

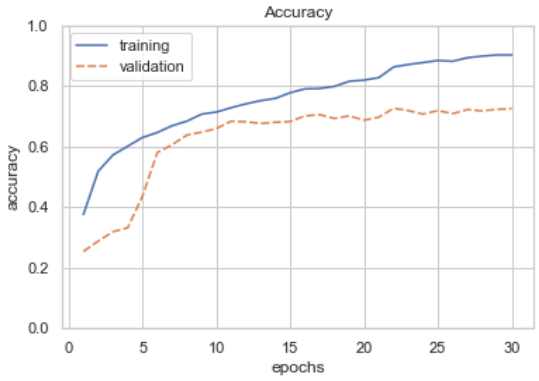

Fig. V.5.12

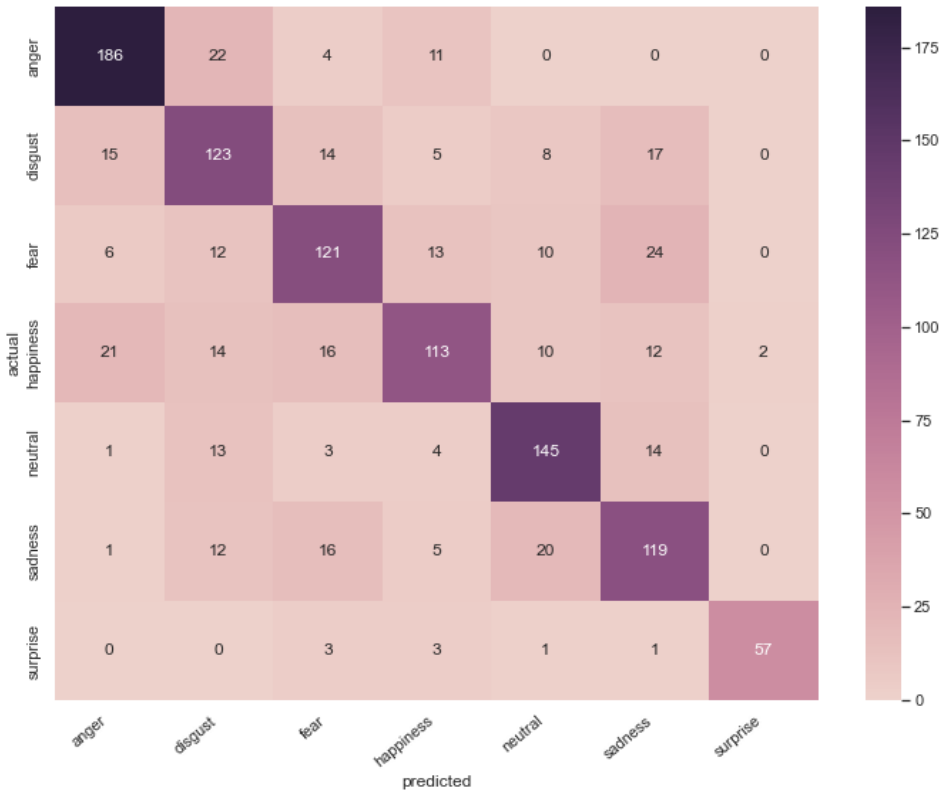

Fig. V.5.13

	class	accuracy	precision	recall	f1_score	sensitivity	specificity
0	anger	0.932331	0.808696	0.834081	0.821192	0.834081	0.045175
1	disgust	0.889724	0.627551	0.675824	0.650794	0.675824	0.071921
2	fear	0.898914	0.683616	0.650538	0.666667	0.650538	0.055391
3	happiness	0.903091	0.733766	0.601064	0.660819	0.601064	0.040634
4	neutral	0.929825	0.747423	0.805556	0.775401	0.805556	0.048181
5	sadness	0.898079	0.636364	0.687861	0.661111	0.687861	0.066406
6	surprise	0.991646	0.966102	0.876923	0.919355	0.876923	0.001767

Fig. V.5.14

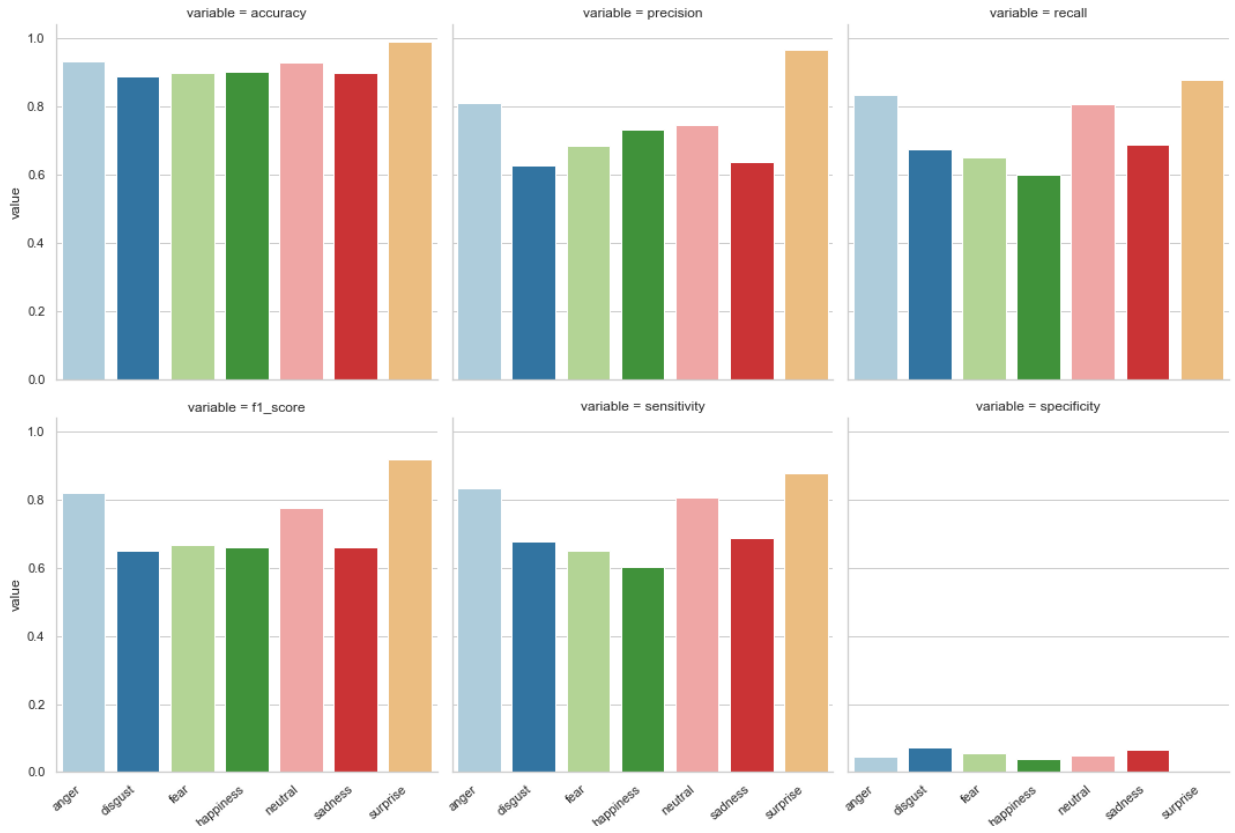

Fig. V.5.15

Set 2 - male samples

Model: Conv1D

Features: Augmented and masked MFCC+Delta

Validation accuracy: 61

Test accuracy: 61

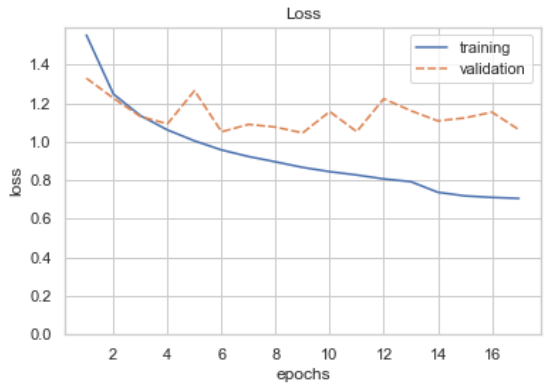

Fig. V.5.16

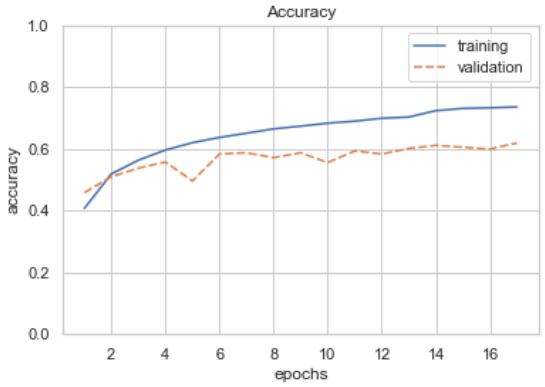

Fig. V.5.17

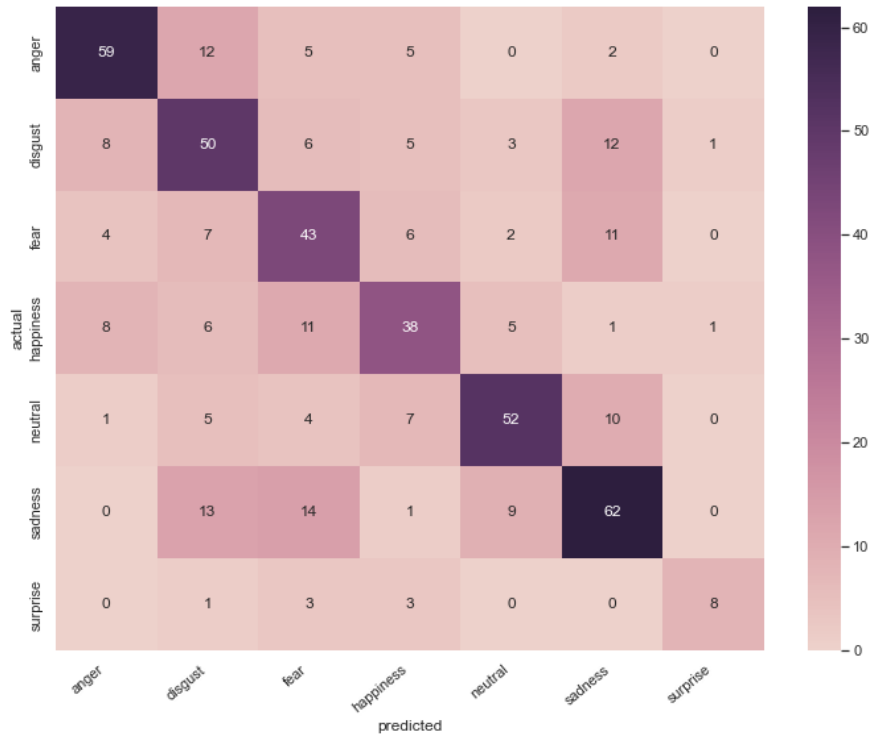

Fig. V.5.18

	class	accuracy	precision	recall	f1_score	sensitivity	specificity
0	anger	0.910714	0.737500	0.710843	0.723926	0.710843	0.049881
1	disgust	0.843254	0.531915	0.588235	0.558659	0.588235	0.105012
2	fear	0.855159	0.500000	0.589041	0.540881	0.589041	0.099768
3	happiness	0.882937	0.584615	0.542857	0.562963	0.542857	0.062212
4	neutral	0.908730	0.732394	0.658228	0.693333	0.658228	0.044706
5	sadness	0.855159	0.632653	0.626263	0.629442	0.626263	0.088889
6	surprise	0.982143	0.800000	0.533333	0.640000	0.533333	0.004090

Fig. V.5.19

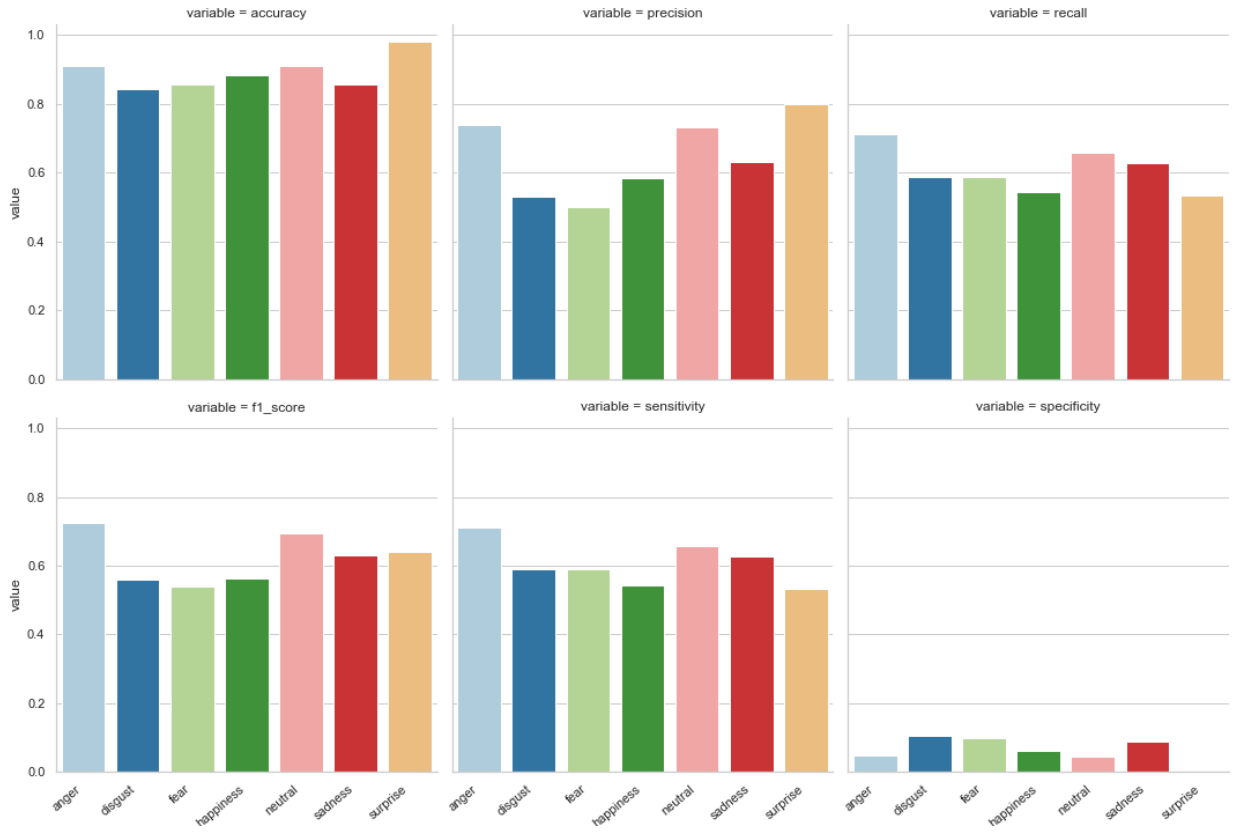

Fig. V.5.20

Model: Conv1D

Features: Augmented Log Mel-Spectrogram

Validation accuracy: 63

Test accuracy: 61

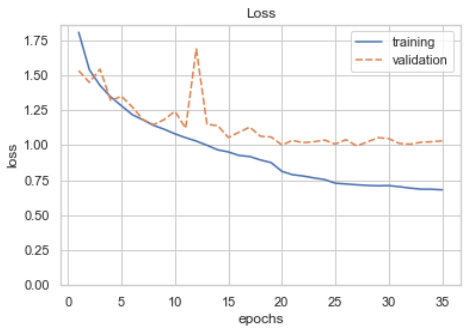

Fig. V.5.21

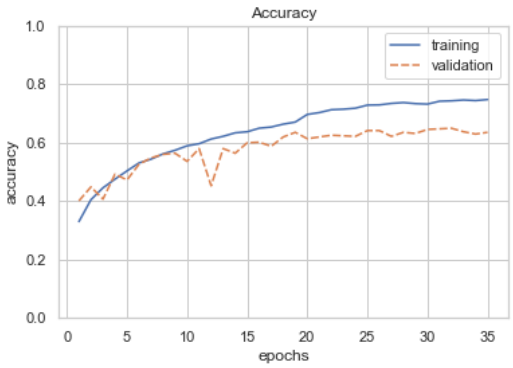

Fig. V.5.22

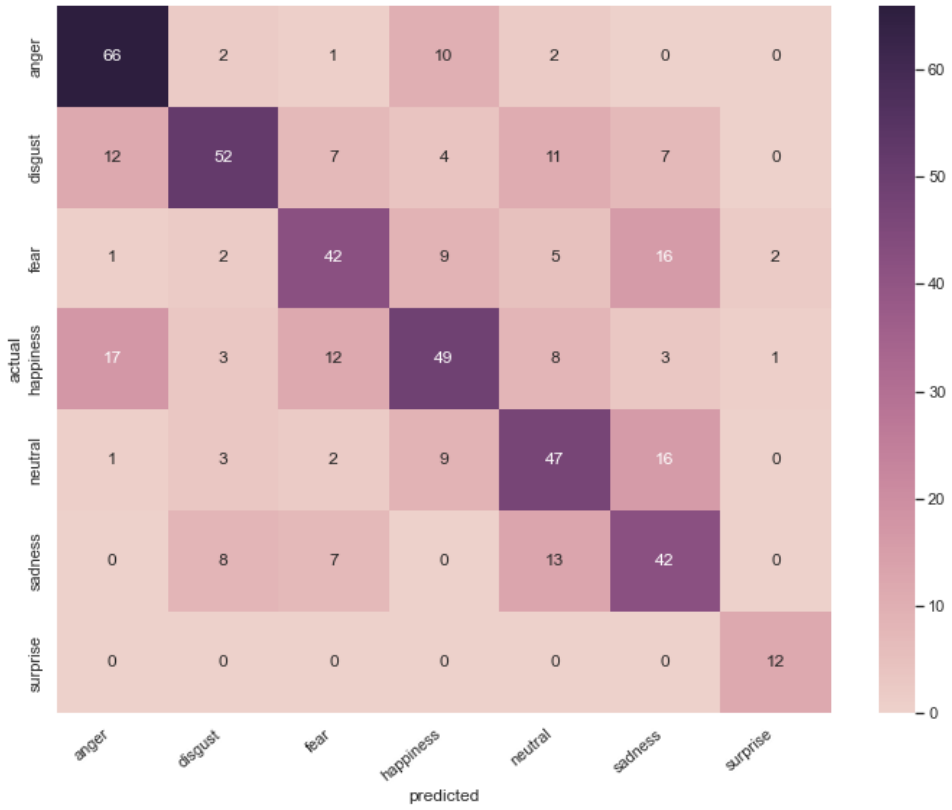

Fig. V.5.23

	class	accuracy	precision	recall	f1_score	sensitivity	specificity
0	anger	0.908730	0.680412	0.814815	0.741573	0.814815	0.073286
1	disgust	0.882937	0.742857	0.559140	0.638037	0.559140	0.043796
2	fear	0.873016	0.591549	0.545455	0.567568	0.545455	0.067916
3	happiness	0.849206	0.604938	0.526882	0.563218	0.526882	0.077859
4	neutral	0.861111	0.546512	0.602564	0.573171	0.602564	0.091549
5	sadness	0.861111	0.500000	0.600000	0.545455	0.600000	0.096774
6	surprise	0.994048	0.800000	1.000000	0.888889	1.000000	0.006098

Fig. V.5.24

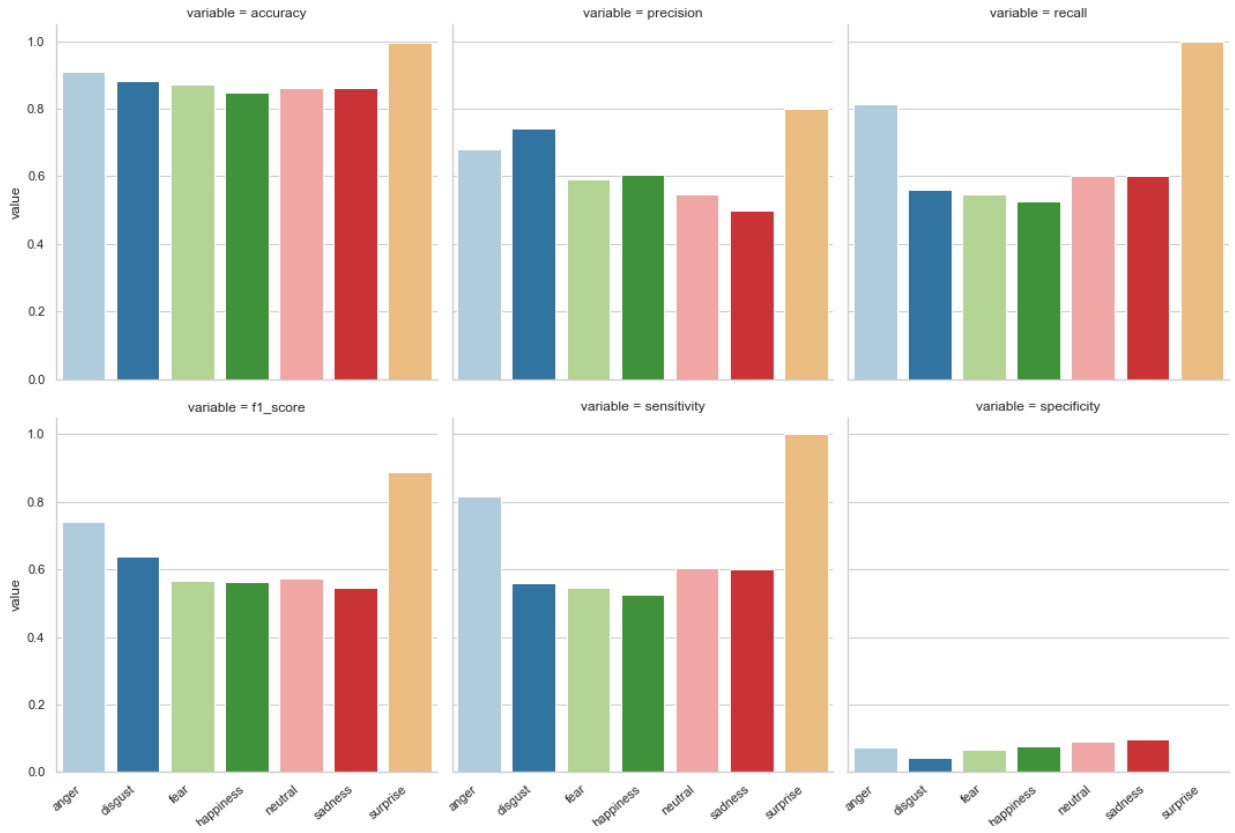

Fig. V.5.25

Model: Conv 2D

Features: Image augmented MFCC

Validation accuracy: 64

Test accuracy: 65

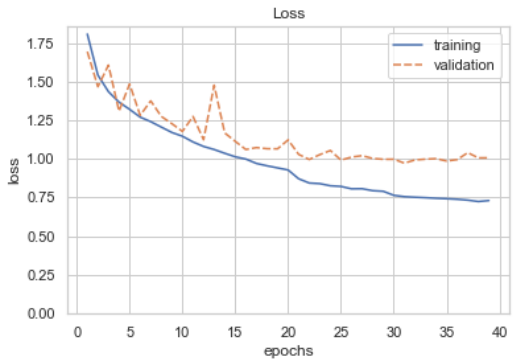

Fig. V.5.26

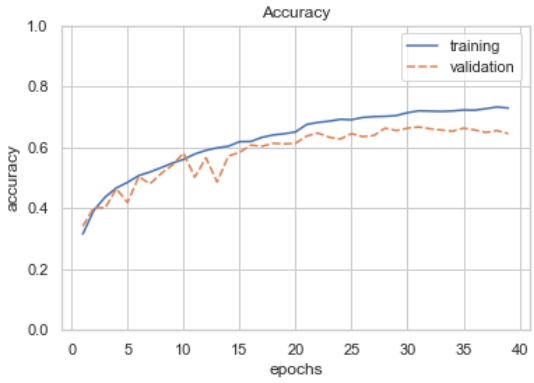

Fig. V.5.27

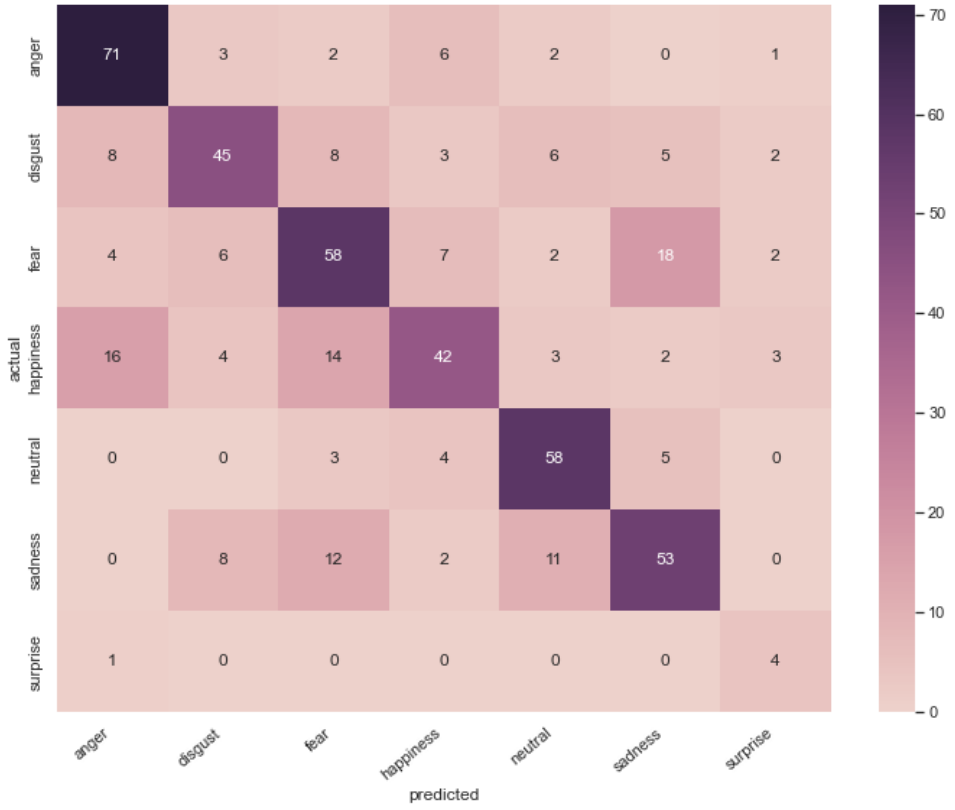

Fig. V.5.28

	class	accuracy	precision	recall	f1_score	sensitivity	specificity
0	anger	0.914683	0.710000	0.835294	0.767568	0.835294	0.069212
1	disgust	0.894841	0.681818	0.584416	0.629371	0.584416	0.049180
2	fear	0.845238	0.597938	0.597938	0.597938	0.597938	0.095823
3	happiness	0.873016	0.656250	0.500000	0.567568	0.500000	0.052381
4	neutral	0.928571	0.707317	0.828571	0.763158	0.828571	0.055300
5	sadness	0.875000	0.638554	0.616279	0.627219	0.616279	0.071770
6	surprise	0.982143	0.333333	0.800000	0.470588	0.800000	0.016032

Fig. V.5.29

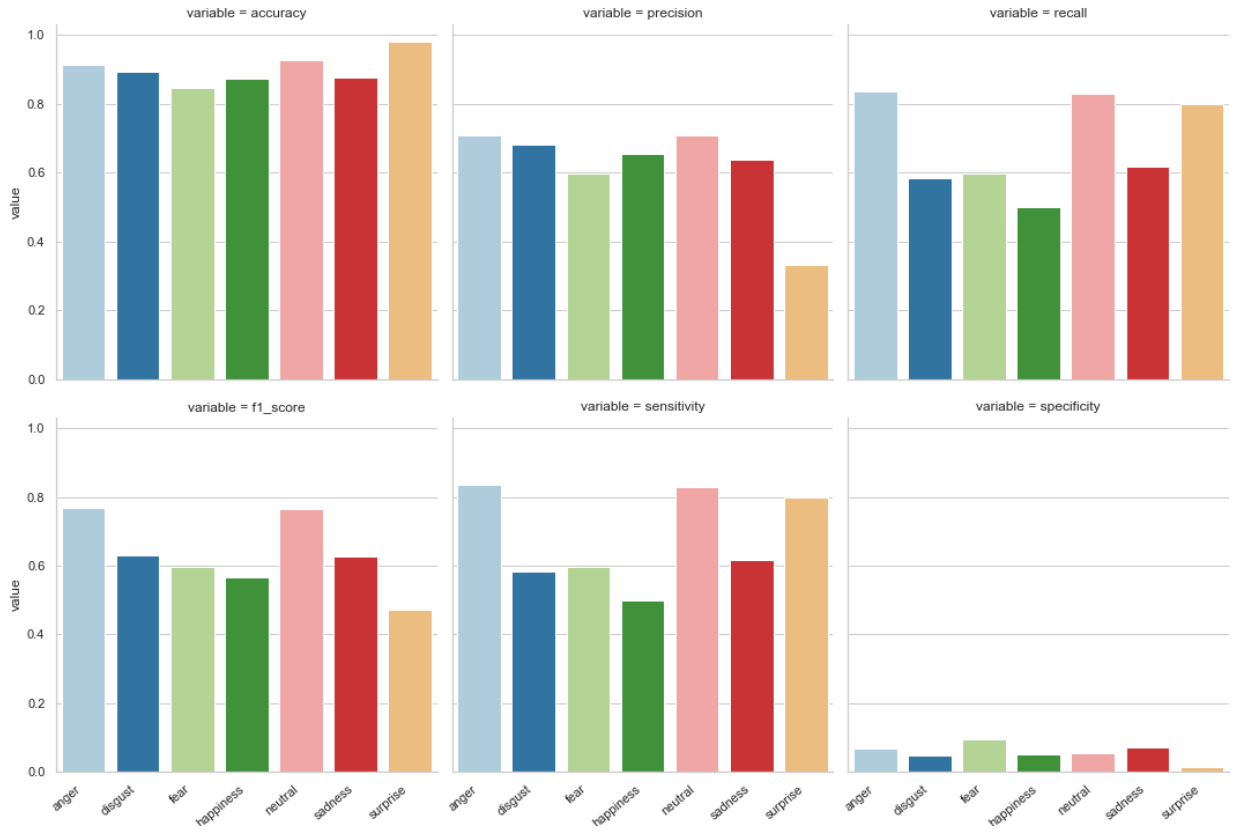

Fig. V.5.30

Set 3 - female samples

Model: Conv 1D

Features: MFCC+RMS+ZCR

Validation accuracy: 78

Test accuracy: 78

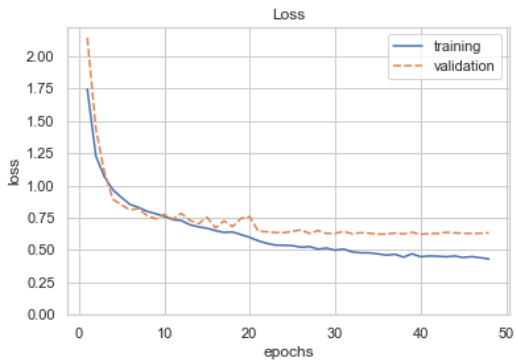

Fig. V.5.31

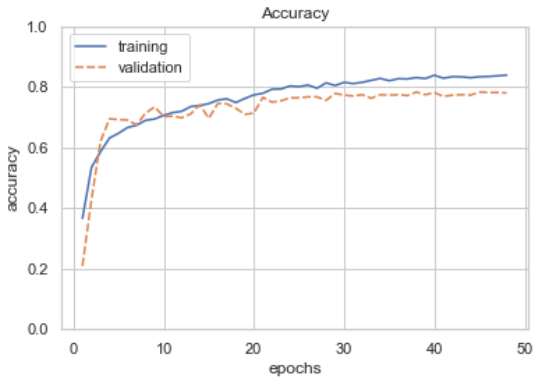

Fig. V.5.32

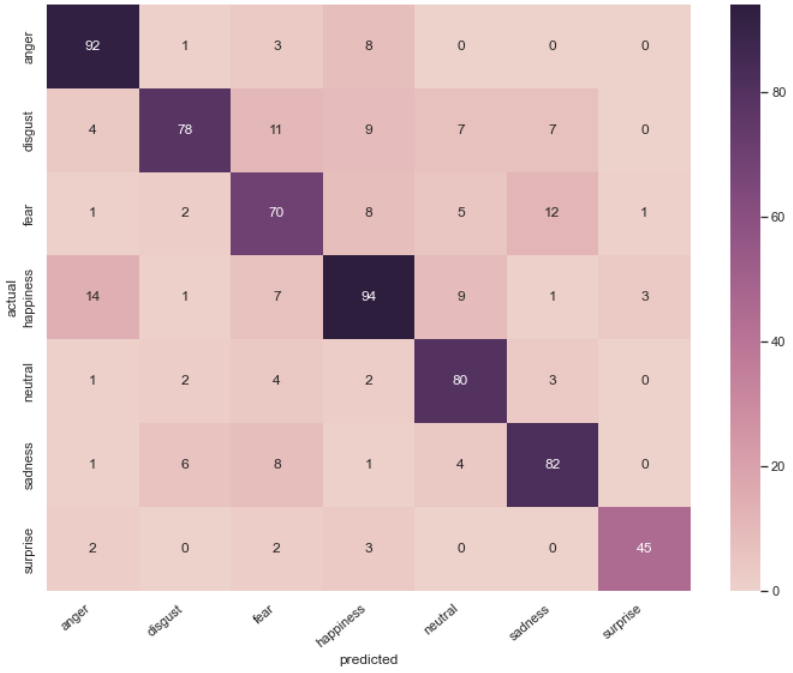

Fig. V.5.33

	class	accuracy	precision	recall	f1_score	sensitivity	specificity
0	anger	0.949568	0.800000	0.884615	0.840183	0.884615	0.038983
1	disgust	0.927954	0.866667	0.672414	0.757282	0.672414	0.020761
2	fear	0.907781	0.666667	0.707071	0.686275	0.707071	0.058824
3	happiness	0.904899	0.752000	0.728682	0.740157	0.728682	0.054867
4	neutral	0.946686	0.761905	0.869565	0.812183	0.869565	0.041528
5	sadness	0.938040	0.780952	0.803922	0.792271	0.803922	0.038851
6	surprise	0.984150	0.918367	0.865385	0.891089	0.865385	0.006231

Fig. V.5.34

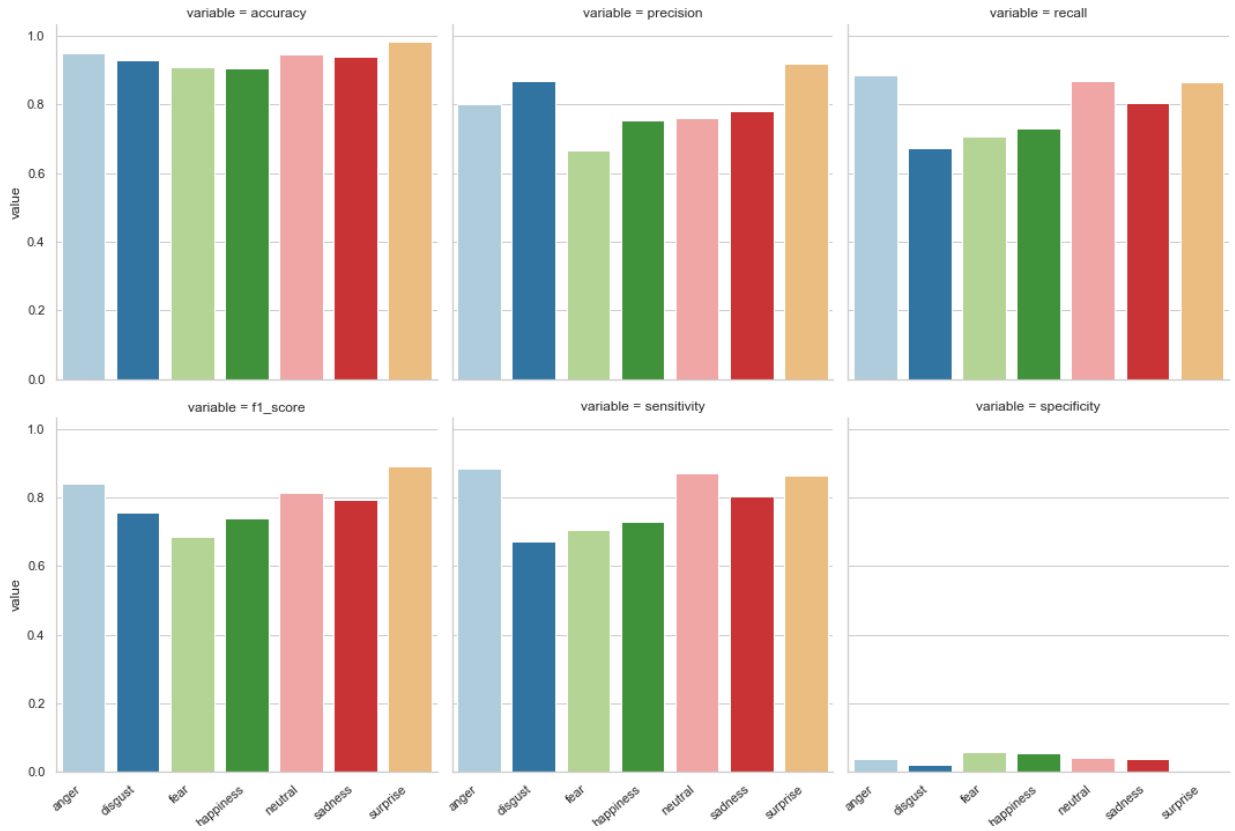

Fig. V.5.35

Model: Conv 1D

Features: Augmented and masked Log Mel-Spectrogram

Validation accuracy: 81

Test accuracy: 81

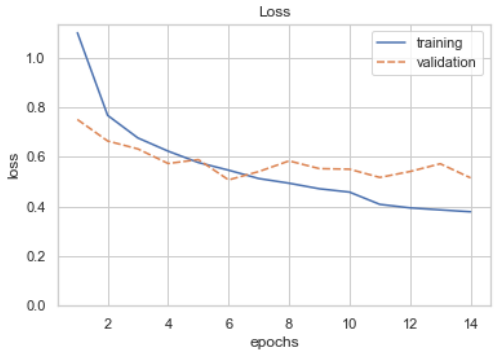

Fig. V.5.36

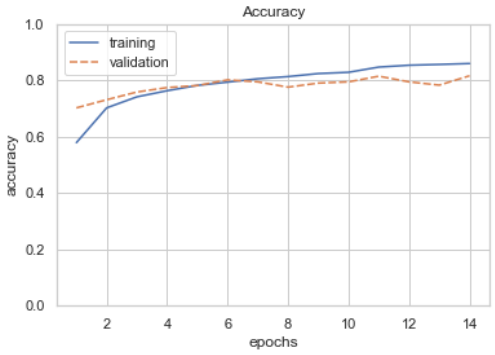

Fig. V.5.37

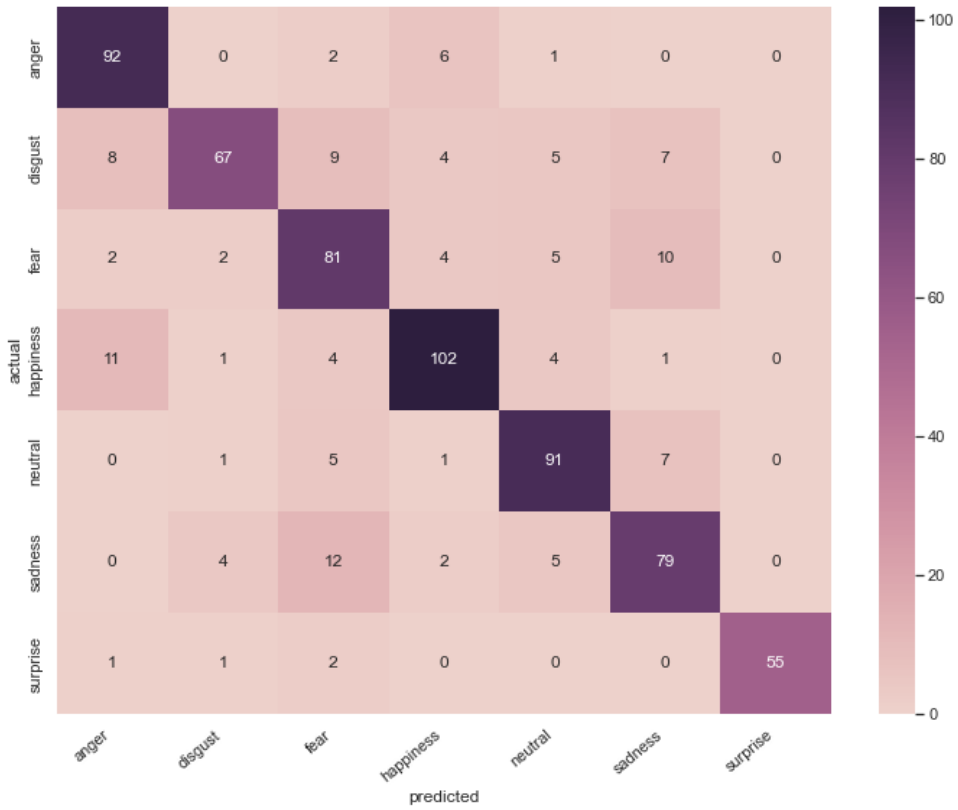

Fig. V.5.38

	class	accuracy	precision	recall	f1_score	sensitivity	specificity
0	anger	0.955331	0.807018	0.910891	0.855814	0.910891	0.037099
1	disgust	0.939481	0.881579	0.670000	0.761364	0.670000	0.015152
2	fear	0.917867	0.704348	0.778846	0.739726	0.778846	0.057627
3	happiness	0.945245	0.857143	0.829268	0.842975	0.829268	0.029772
4	neutral	0.951009	0.819820	0.866667	0.842593	0.866667	0.033956
5	sadness	0.930836	0.759615	0.774510	0.766990	0.774510	0.042230
6	surprise	0.994236	1.000000	0.932203	0.964912	0.932203	0.000000

Fig. V.5.39

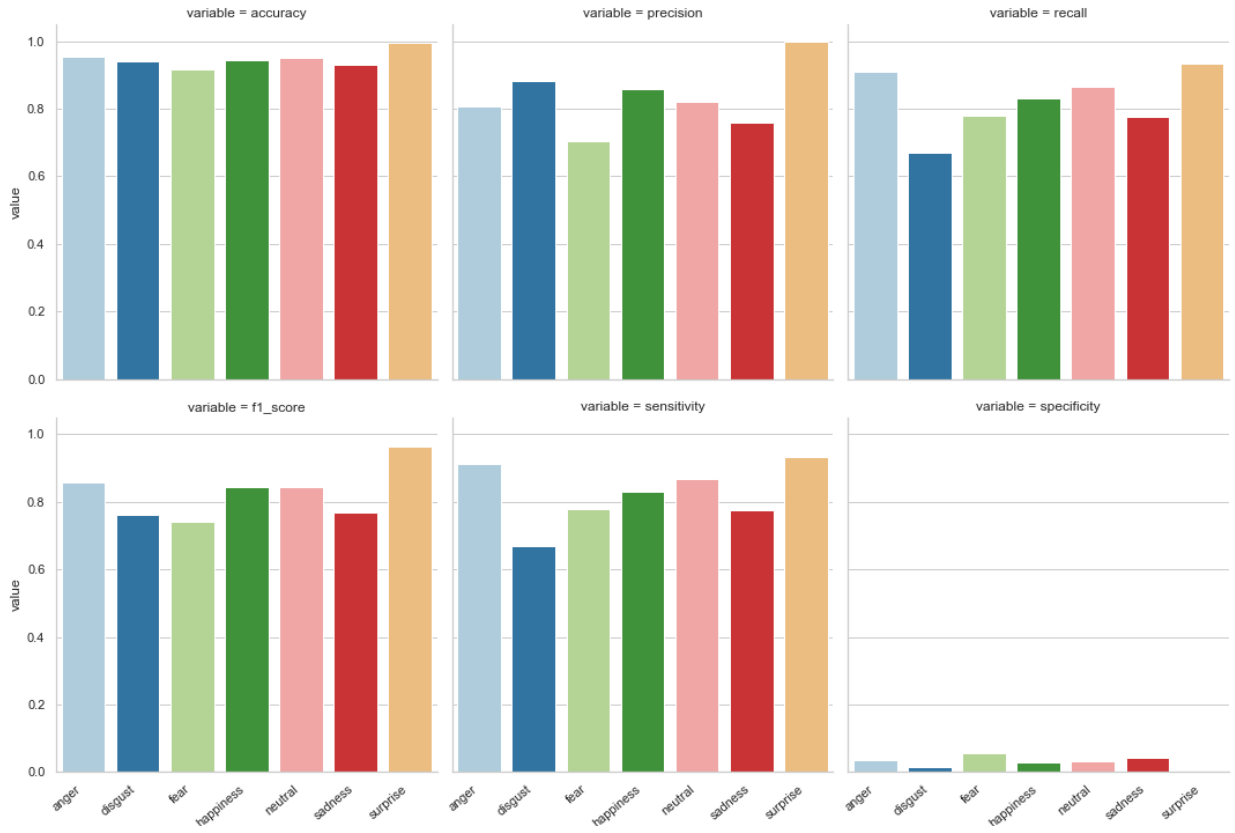

Fig. V.5.40

Model: Conv 2D

Features: Image augmented Log Mel-Spectrogram

Validation accuracy: 79

Test accuracy: 79

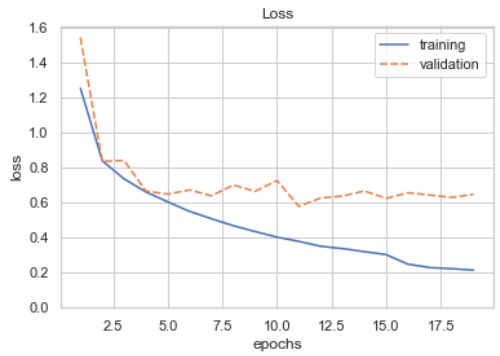

Fig. V.5.41

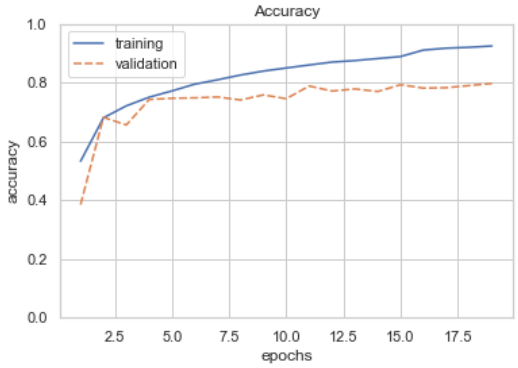

Fig. V.5.42

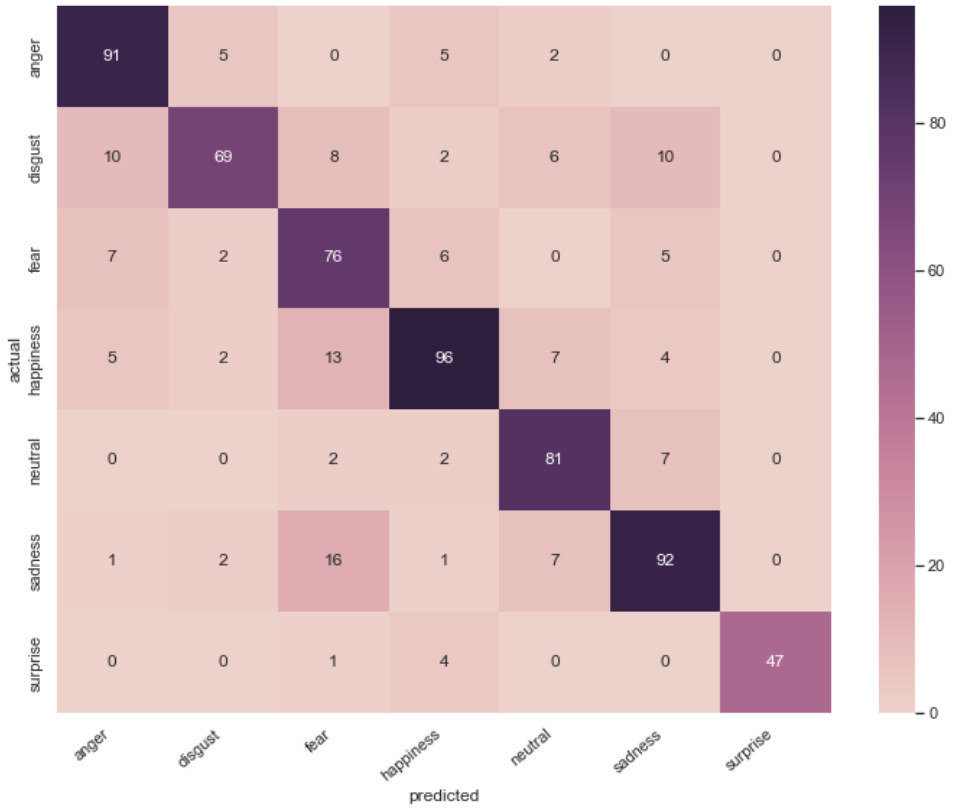

Fig. V.5.43

	class	accuracy	precision	recall	f1_score	sensitivity	specificity
0	anger	0.949568	0.798246	0.883495	0.838710	0.883495	0.038917
1	disgust	0.932277	0.862500	0.657143	0.745946	0.657143	0.018676
2	fear	0.913545	0.655172	0.791667	0.716981	0.791667	0.066890
3	happiness	0.926513	0.827586	0.755906	0.790123	0.755906	0.035273
4	neutral	0.952450	0.786408	0.880435	0.830769	0.880435	0.036545
5	sadness	0.923631	0.779661	0.773109	0.776371	0.773109	0.045217
6	surprise	0.992795	1.000000	0.903846	0.949495	0.903846	0.000000

Fig. V.5.44

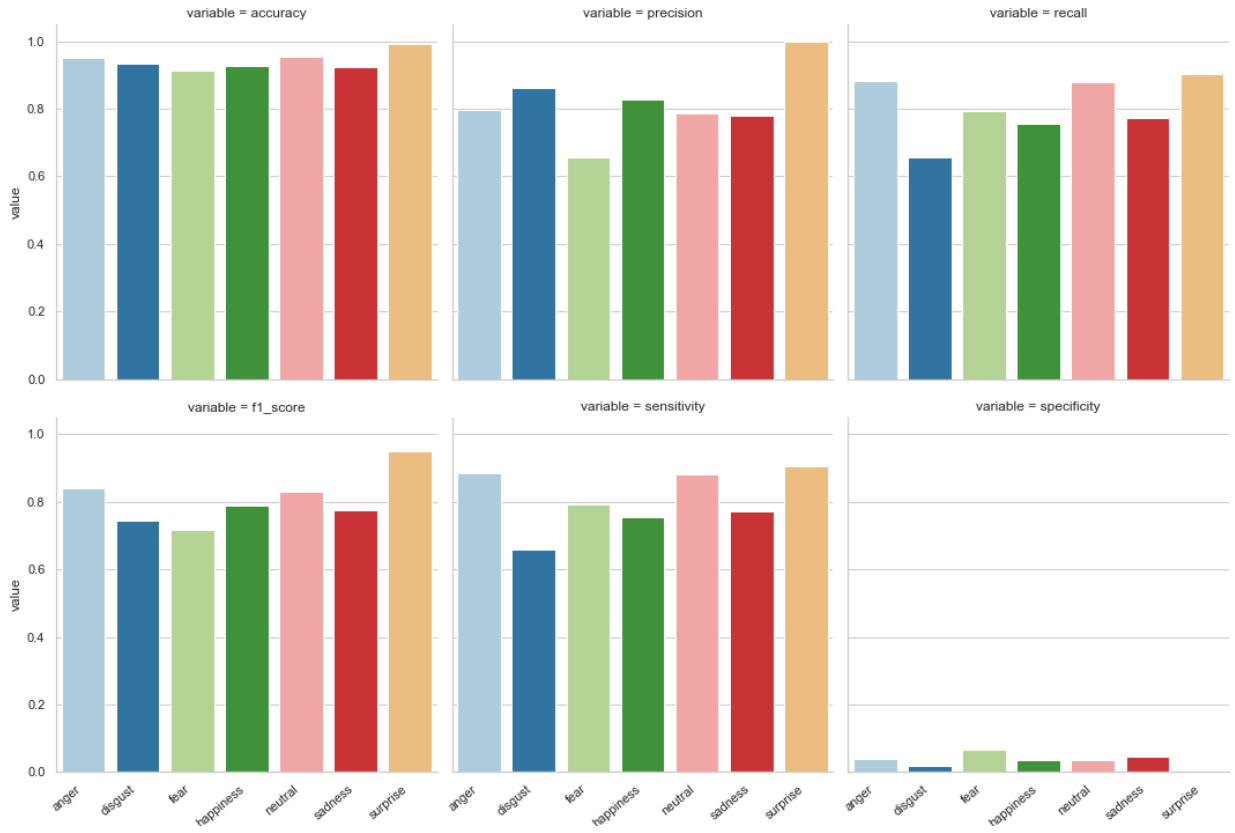

Fig. V.5.45

Set 4 - all samples, 14 labels

Model: Conv 1D

Features: Augmented MFCC+Delta

Validation accuracy: 68

Test accuracy: 71

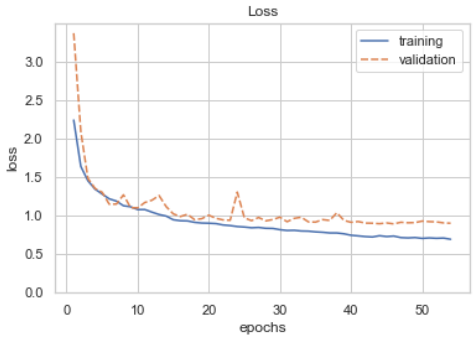

Fig. V.5.46

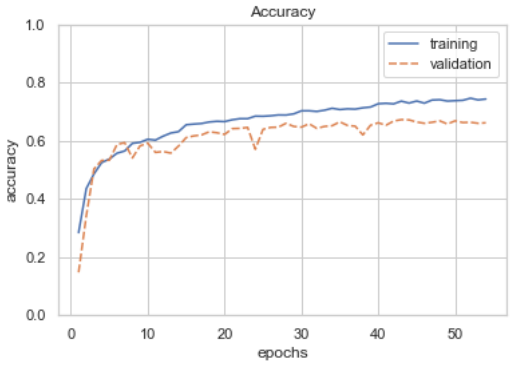

Fig. V.5.47

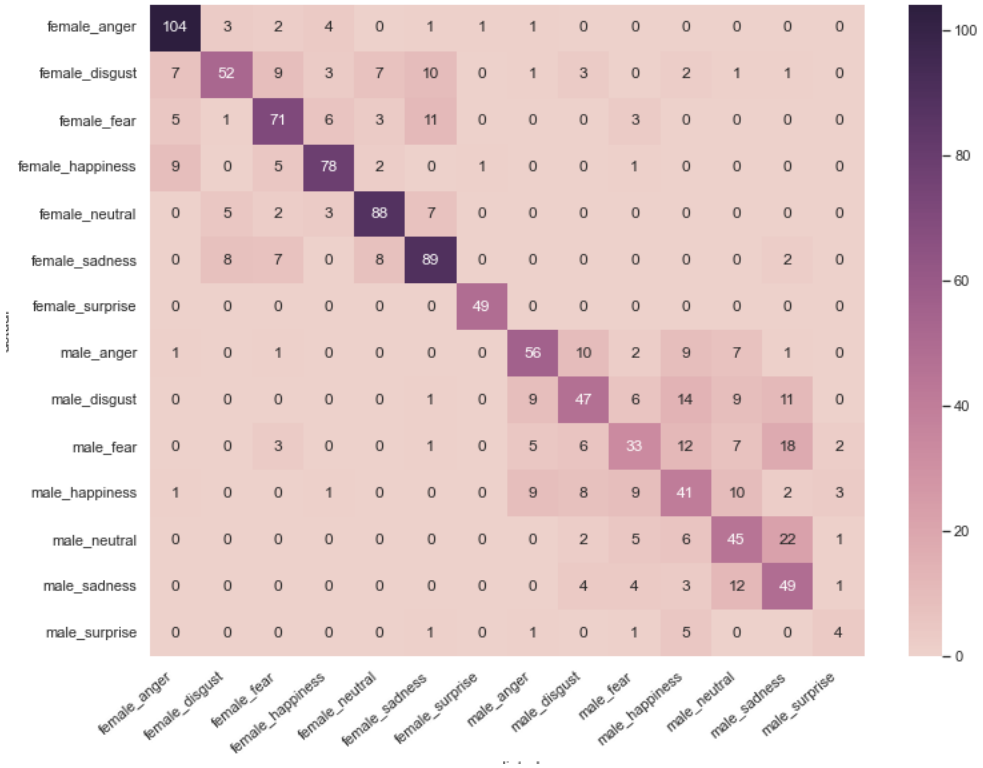

Fig. V.5.48

	class	accuracy	precision	recall	f1_score	sensitivity	specificity
0	female_anger	0.970760	0.818898	0.896552	0.855967	0.896552	0.021277
1	female_disgust	0.949039	0.753623	0.541667	0.630303	0.541667	0.015441
2	female_fear	0.951546	0.710000	0.710000	0.710000	0.710000	0.026436
3	female_happiness	0.970760	0.821053	0.812500	0.816754	0.812500	0.015441
4	female_neutral	0.969089	0.814815	0.838095	0.826291	0.838095	0.018315
5	female_sadness	0.952381	0.735537	0.780702	0.757447	0.780702	0.029548
6	female_surprise	0.998329	0.960784	1.000000	0.980000	1.000000	0.001742
7	male_anger	0.952381	0.682927	0.643678	0.662722	0.643678	0.023423
8	male_disgust	0.930660	0.587500	0.484536	0.531073	0.484536	0.030000
9	male_fear	0.928989	0.515625	0.379310	0.437086	0.379310	0.027928
10	male_happiness	0.921470	0.445652	0.488095	0.465909	0.488095	0.045822
11	male_neutral	0.931495	0.494505	0.555556	0.523256	0.555556	0.041219
12	male_sadness	0.932331	0.462264	0.671233	0.547486	0.671233	0.050712
13	male_surprise	0.987469	0.363636	0.333333	0.347826	0.333333	0.005907

Fig. V.5.49

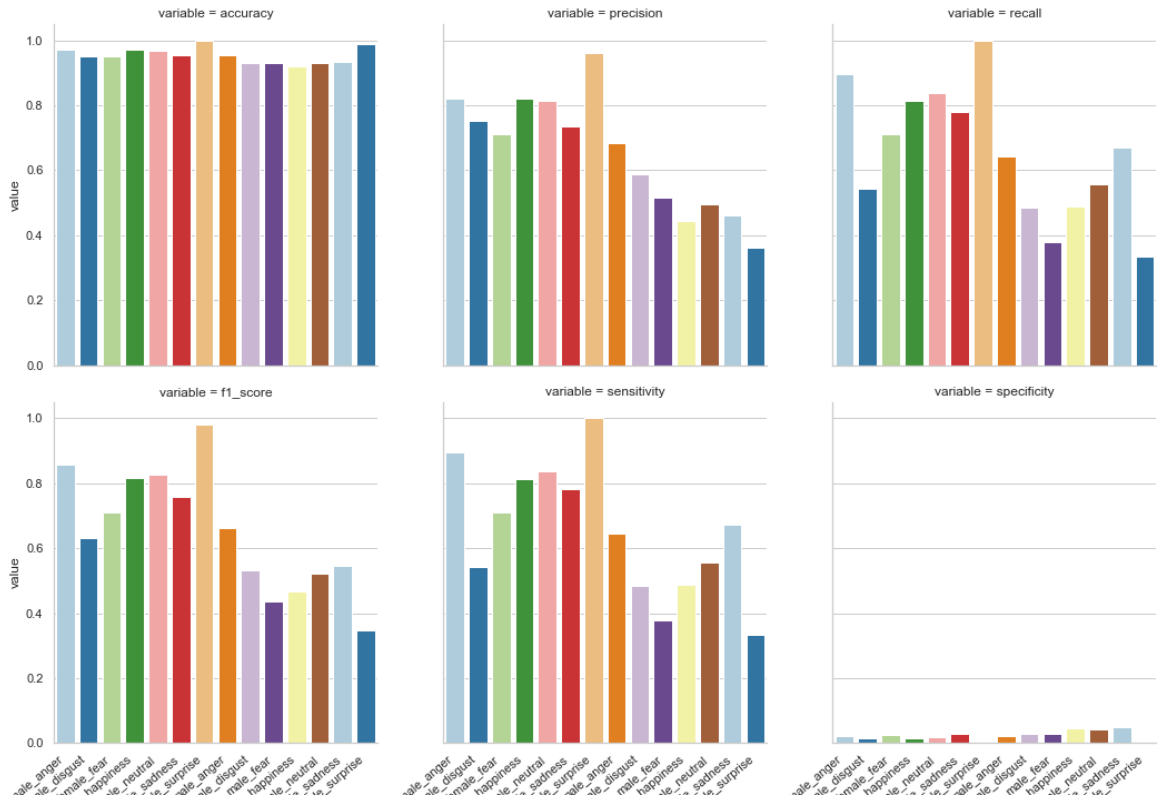

Fig. V.5.50

Model: Conv 1D

Features: Augmented and masked Log Mel-Spectrogram

Validation accuracy: 68

Test accuracy: 73

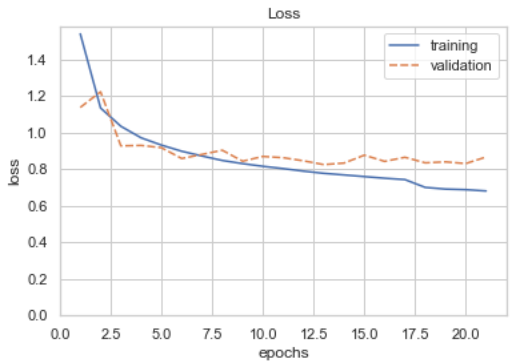

Fig. V.5.51

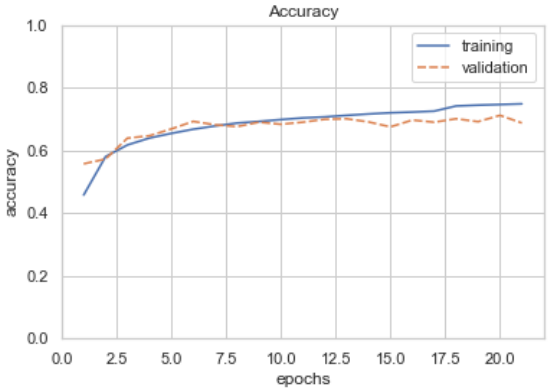

Fig. V.5.52

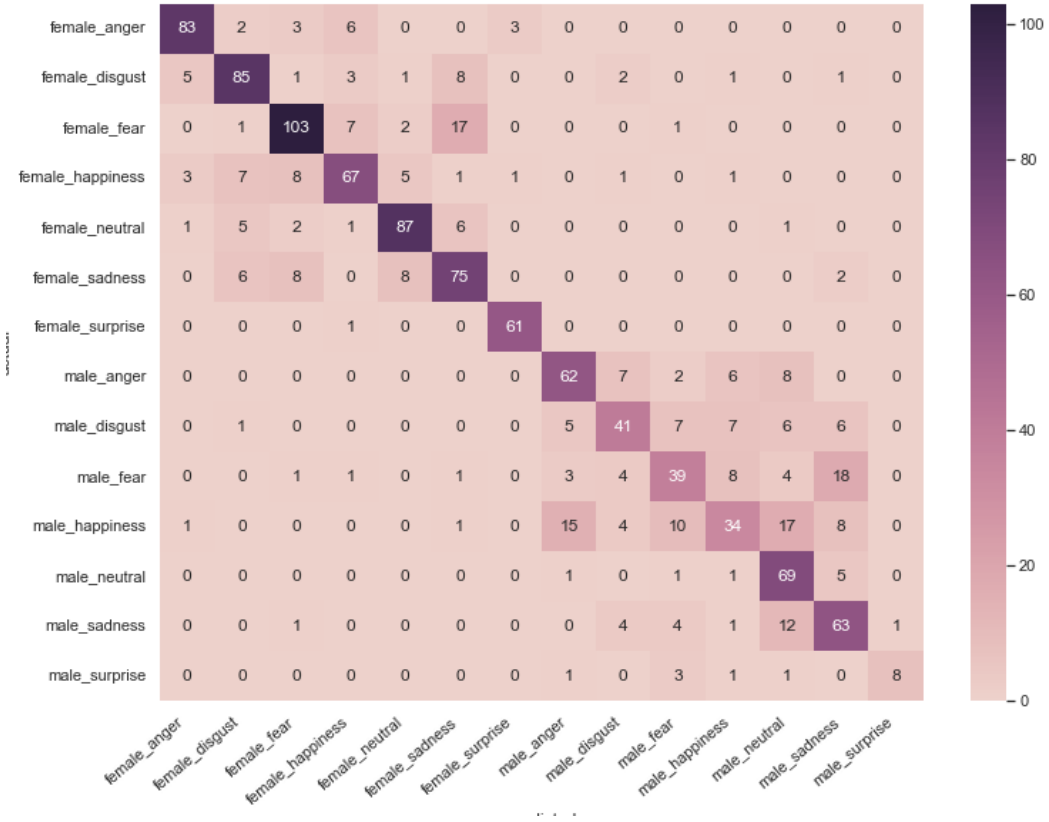

Fig. V.5.53

	class	accuracy	precision	recall	f1_score	sensitivity	specificity
0	female_anger	0.979950	0.892473	0.855670	0.873684	0.855670	0.009091
1	female_disgust	0.963241	0.794393	0.794393	0.794393	0.794393	0.020183
2	female_fear	0.956558	0.811024	0.786260	0.798450	0.786260	0.022514
3	female_happiness	0.961571	0.779070	0.712766	0.744444	0.712766	0.017226
4	female_neutral	0.973266	0.844660	0.844660	0.844660	0.844660	0.014625
5	female_sadness	0.951546	0.688073	0.757576	0.721154	0.757576	0.030965
6	female_surprise	0.995823	0.938462	0.983871	0.960630	0.983871	0.003524
7	male_anger	0.959900	0.712644	0.729412	0.720930	0.729412	0.022482
8	male_disgust	0.954887	0.650794	0.561644	0.602941	0.561644	0.019573
9	male_fear	0.943191	0.582090	0.493671	0.534247	0.493671	0.025045
10	male_happiness	0.931495	0.566667	0.377778	0.453333	0.377778	0.023487
11	male_neutral	0.952381	0.584746	0.896104	0.707692	0.896104	0.043750
12	male_sadness	0.947368	0.611650	0.732558	0.666667	0.732558	0.036004
13	male_surprise	0.994152	0.888889	0.571429	0.695652	0.571429	0.000845

Fig. V.5.54

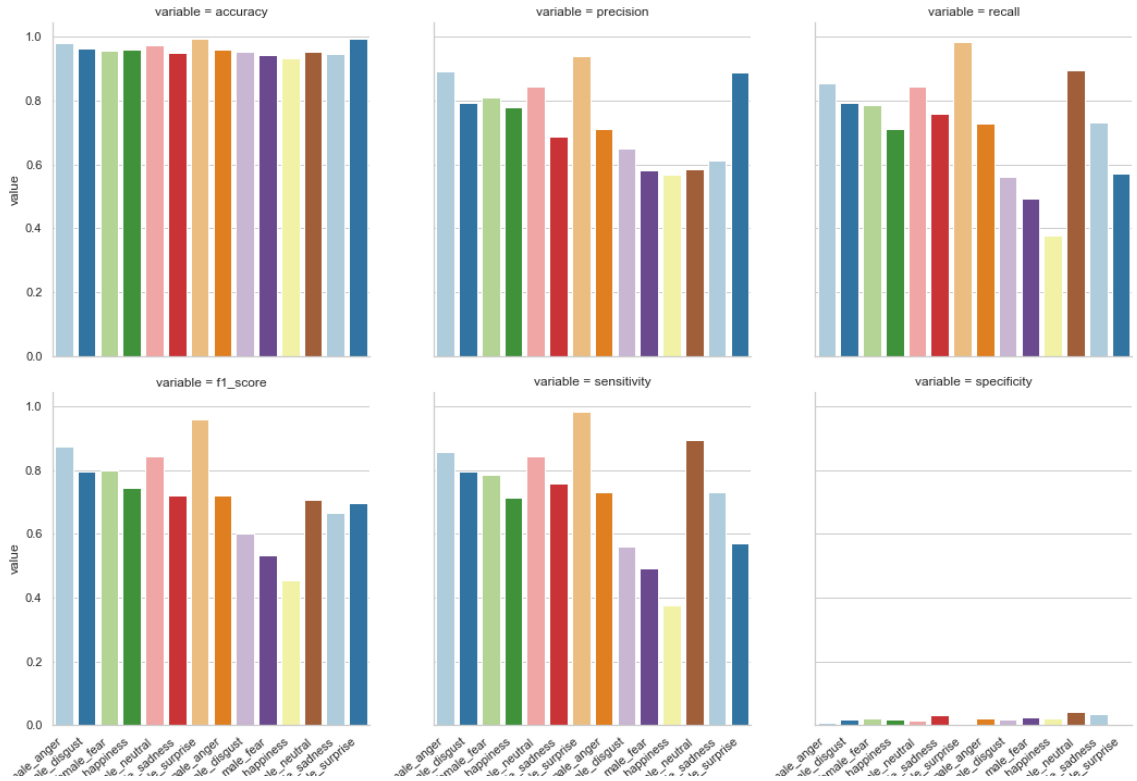

Fig. V.5.55

Model: Conv 2D

Features: Image augmented MFCC

Validation accuracy: 71

Test accuracy: 74

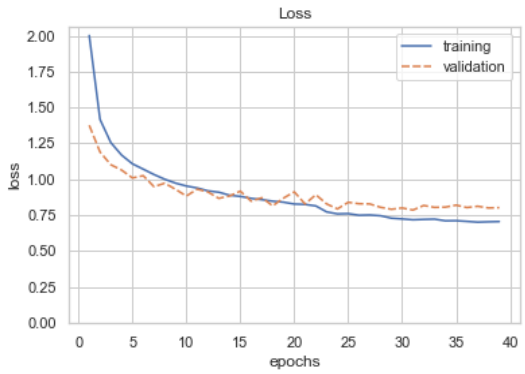

Fig. V.5.56

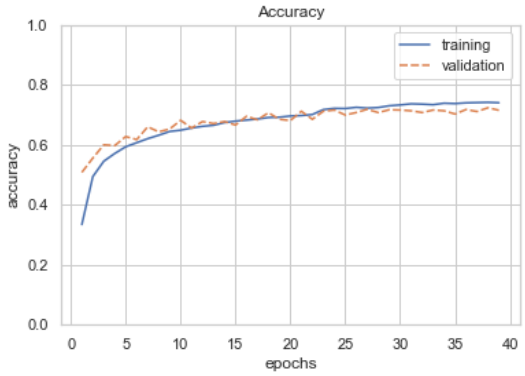

Fig. V.5.57

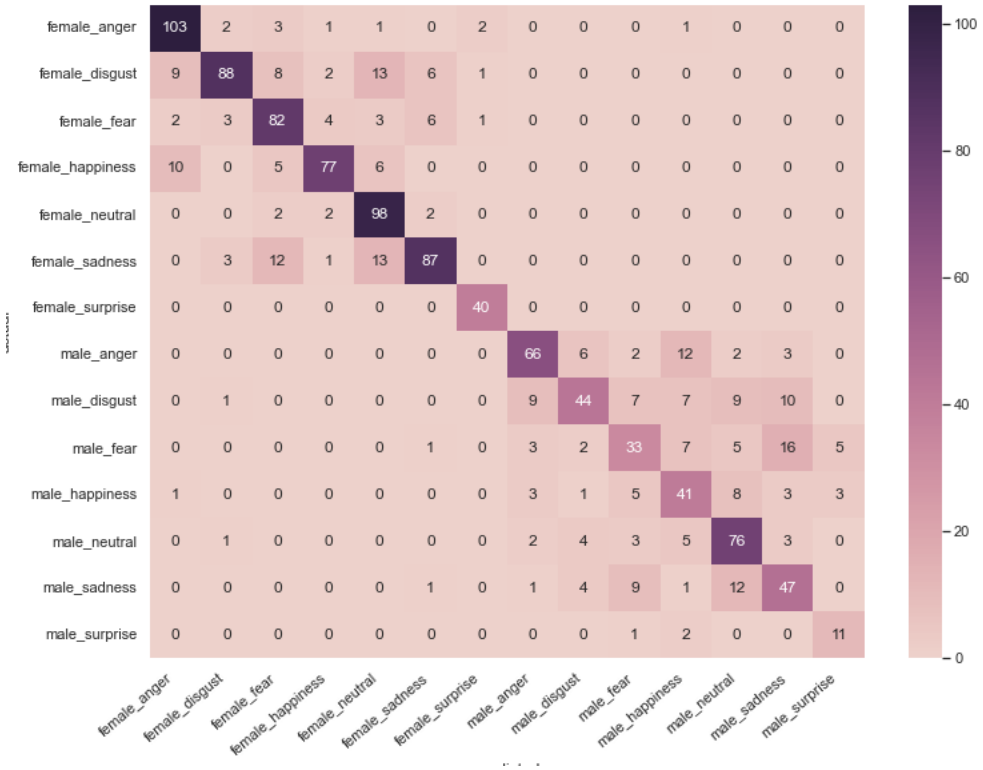

Fig. V.5.58

	class	accuracy	precision	recall	f1_score	sensitivity	specificity
0	female_anger	0.973266	0.824000	0.911504	0.865546	0.911504	0.020295
1	female_disgust	0.959064	0.897959	0.692913	0.782222	0.692913	0.009346
2	female_fear	0.959064	0.732143	0.811881	0.769953	0.811881	0.027372
3	female_happiness	0.974102	0.885057	0.785714	0.832432	0.785714	0.009099
4	female_neutral	0.964912	0.731343	0.942308	0.823529	0.942308	0.032937
5	female_sadness	0.962406	0.844660	0.750000	0.794521	0.750000	0.014801
6	female_surprise	0.996658	0.909091	1.000000	0.952381	1.000000	0.003457
7	male_anger	0.964077	0.785714	0.725275	0.754286	0.725275	0.016275
8	male_disgust	0.949875	0.721311	0.505747	0.594595	0.505747	0.015315
9	male_fear	0.944862	0.550000	0.458333	0.500000	0.458333	0.024000
10	male_happiness	0.950710	0.539474	0.630769	0.581560	0.630769	0.030919
11	male_neutral	0.954887	0.678571	0.808511	0.737864	0.808511	0.032638
12	male_sadness	0.947368	0.573171	0.626667	0.598726	0.626667	0.031194
13	male_surprise	0.990810	0.578947	0.785714	0.666667	0.785714	0.006762

Fig. V.5.59

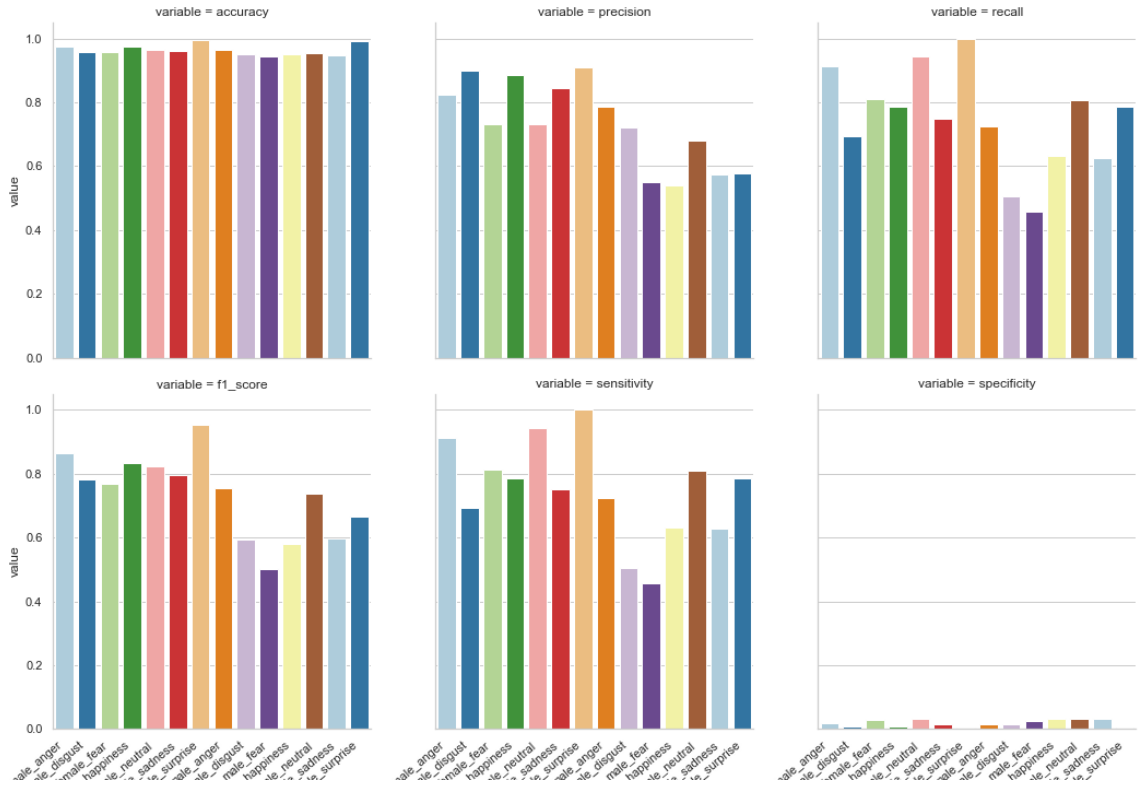

Fig. V.5.60

VI. Related work

As stated in this paper published in 2019, “Emotion recognition from Speech”, audio samples of speech contain key features in extracting information in a structured way and the extracted features are the most important factor for an emotion recognition system’s performance. [18]

The main steps in creating such a system are the same for this project as they are in the paper : selecting data, processing data, extracting features, creating models and comparing them. The methods used for this project differ for the most part from the ones used in the paper but there are a few similarities.

In this paper, they selected the RAVDESS dataset which is one of the four selected original datasets for this project. The labels were chosen to be based on both sex and emotion which again is the same for one of the four datasets created for this project.

The data was first trimmed and various techniques for cleaning the signal were tried including: filtering, voice-activity detection, spectral subtraction. Wiener filtering was implemented in the end. In this project there were no attempts to clean the signal as I considered it unnecessary after looking at the visualization of the samples..

For feature extraction the MFCCs were extracted along with the Delta and the Delta-Delta and different models were trained in three configurations : one using the MFCC as features, another using the MFCC and Delta as features and the last one using the MFCC and the Delta-Delta as features. This is another similarity this project shares with the paper as the MFCC and the Deltas were also some of the feature extraction methods used in this project, but the configurations were different.

The second feature extraction method used in this paper is the Mel-Spectrogram. This method was also used in this project, but again in different configurations.

For the models they considered a one dimensional CNN for the raw data, a two dimensional CNN for the data gathered using the feature extraction models mentioned above and a three dimensional CNN for the image representation of the MFCCs and the Mel spectrogram.

Two dimensional CNN-LSTM were also trained on the extracted data as well as a one dimensional CNN-LSTM for the raw audio.

They implemented two types of Hidden Markov Models as well since they were widely used in speech recognition tasks before Recurrent Neural Networks. The two types implemented were a Gausian HMM and a GMM HMM.

The RAVDESS dataset was used for creating the train, validation and test sets but the TESS dataset was also added to the train set in order to test the model on a completely different dataset but with similar voices.

As we can see in the Fig. VI.1. , representing a summary of their results, they are very similar to the results of this project. Some combinations of feature extraction and architecture performed slightly differently since the methods used were not the same, the bet validation accuracy in both cases was around the 70% mark, one model used in the paper being able to reach an accuracy of 90% only when it was used on a binary classification of the data into positive or negative emotions.

Features	Architecture	No. of Class	Validation Acc.	Test Acc.
Pure Audio	1D CNN + LSTM	12	61.6%	48.8%
Log Mel Spectogram	4 Layer 2D CNN	12	70.31%	65%
29 Coefficients:MFCC+Delta	1 Layer 2D CNN	12	56%	53%
Log Mel Spectogram	HMM	12	-	31.25%
Log Mel Spectogram	3 Layer 3D CNN	12	66%	55%
Log Mel Spectogram	**2D CNN with Global Avg. Pool**	**14**	**70%**	**66%**
Log Mel Spectogram	**2D CNN with Global Avg. Pool**	**2**	**90%**	**86%**

Fig. VI.1. Summary of results in the paper

For comparison purposes Fig. VI.2. shows the visualization of the confusion matrix of the best performing model on 14 classes from the paper.

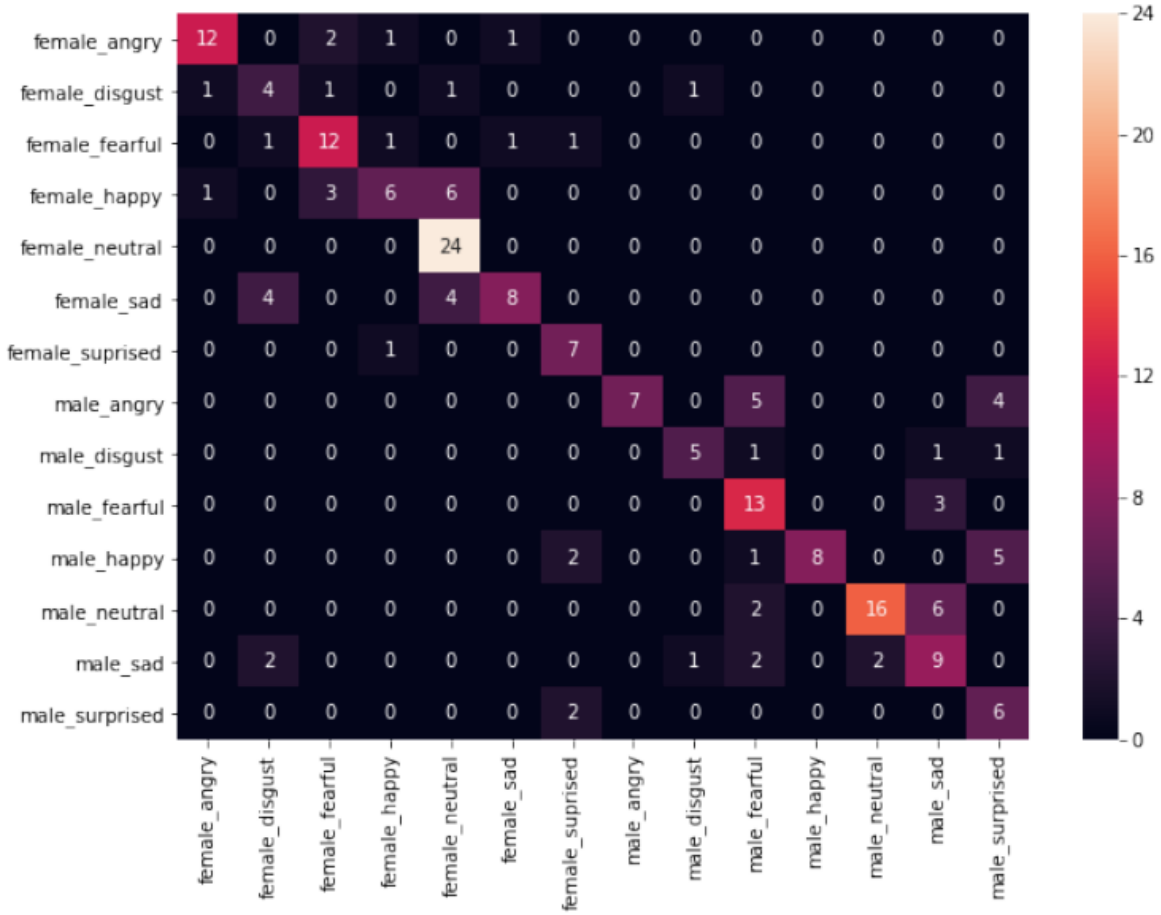

Fig. VI.2. - Confusion matrix

VII. Conclusions

The series of experiments conducted during the implementation of the code backing this paper revealed interesting results. The results should prove useful in the future when selecting a model for a similar task involving emotion recognition based on sound. The methods presented here can also be used in compound tasks such as emotion recognition based on video data or even further, in sentient artificial intelligence agents or complete AI's.

One of the main factors in determining the quality of the output model was the preprocessing applied to the signals and their resulting coefficients. Without these, regular neural networks would have a hard time classifying the wide spectrum of emotion of a human being.

As the results show, neural networks are capable of processing sound signals transformed in different ways, or their coefficients, resulting in an agent that can, to a degree, recognize human emotions.

All configurations performed better on female voices as input, even when both the female and male samples came from the same original dataset. This could be because emotion is more easily recognizable in higher voices or simply because women express emotions more clearly when they speak.

The goal of the project was achieved by obtaining and comparing multiple machine learning and sound processing techniques to a degree where they can be chosen depending on the needs of one's problem.

VIII. Bibliography

[1] Librosa. (n.d.). Librosa Documentation. Retrieved May 30, 2022, from http://librosa.org/doc/main/effects.html

[2] *Tensorflow*. TensorFlow. (n.d.). Retrieved May 30, 2022, from https://www.tensorflow.org/

[3] Park, D. S., Chan, W., Zhang, Y., Chiu, C. C., Zoph, B., Cubuk, E. D., & Le, Q. V. (2019). Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*.

[4] Douglas O'Shaughnessy (1987). Speech communication: human and machine. Addison-Wesley. p. 150. ISBN 978-0-201-16520-3.

[5] Sahidullah, Md.; Saha, Goutam (May 2012). "Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition". Speech Communication. 54 (4): 543–565. doi:10.1016/j.specom.2011.11.004.

[6] T. Ganchev, N. Fakotakis, and G. Kokkinakis (2005), "Comparative evaluation of various MFCC implementations on the speaker verification task Archived 2011-07-17 at the Wayback Machine," in 10th International Conference on Speech and Computer (SPECOM 2005), Vol. 1, pp. 191–194.

[7] Min Xu; et al. (2004). "HMM-based audio keyword generation" (PDF). In Kiyoharu Aizawa; Yuichi Nakamura; Shin'ichi Satoh (eds.). Advances in Multimedia Information Processing – PCM 2004: 5th Pacific Rim Conference on Multimedia. Springer. ISBN 978-3-540-23985-7.

[8] Bäckström, T. (2019, April 16). *Deltas and Delta-Deltas*. Aalto University Wiki. Retrieved June 2, 2022, from https://wiki.aalto.fi/display/ITSP/Deltas+and+Delta-deltas

[9] Gouyon, F., Pachet, F., & Delerue, O. (2000, December). On the use of zero-crossing rate for an application of classification of percussive sounds. In *Proceedings of the COST G-6 conference on Digital Audio Effects (DAFX-00), Verona, Italy*

[10] IBM Cloud Education. (n.d.). *What are neural networks?* IBM. Retrieved June 3, 2022, from https://www.ibm.com/cloud/learn/neural-networks

[11] Amidi, A., & Amidi, S. (n.d.). *Convolutional Neural Networks cheatsheet star*. CS 230 - Convolutional Neural Networks Cheatsheet. Retrieved May 30, 2022, from https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

[12] IBM Cloud Education. (n.d.). *What are convolutional neural networks?* IBM. Retrieved June 2, 2022, from https://www.ibm.com/cloud/learn/convolutional-neural-networks

[13] Sak, Hasim; Senior, Andrew; Beaufays, Francoise (2014). "Long Short-Term Memory recurrent neural network architectures for large scale acoustic modeling" (PDF).

[14] CheyneyComputerScience. (n.d.). *CheyneyComputerScience/Crema-D: Crowd sourced emotional multimodal actors dataset (crema-D)*. GitHub. Retrieved May 31, 2022, from https://github.com/CheyneyComputerScience/CREMA-D

[15] Livingstone, S. R. (2019, January 19). *Ravdess emotional speech audio*. Kaggle. Retrieved April 29, 2022, from https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio

[16] TSpace. (n.d.). Retrieved June 3, 2022, from https://tspace.library.utoronto.ca/handle/1807/24487

[17] Surrey Audio-visual expressed emotion (SAVEE) database. (n.d.). Retrieved May 25, 2022, from http://kahlan.eps.surrey.ac.uk/savee/Download.html

[18] Venkataramanan, K., & Rajamohan, H. R. (2019). Emotion recognition from speech. *arXiv preprint arXiv:1912.10458*.

[19] Singh, M., & Fang, Y. (2020). Emotion recognition in audio and video using deep neural networks. *arXiv preprint arXiv:2006.08129*.

[20] Ragheb, W., Mirzapour, M., Delfardi, A., Jacquenet, H., & Carbon, L. (2022). Emotional Speech Recognition with Pre-trained Deep Visual Models. *arXiv preprint arXiv:2204.03561*.