Universiteit Leiden
The Netherlands

Opleiding Informatica

Extracting characteristics of emotions from audio

Job van Dijke

Supervisors:
Joost Broekens
Peter van der Putten

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

26/08/2022

Abstract

In this bachelor thesis, we extract different features from audio samples that contain voices in different emotions. These features include melfrequency spectra and MFCCs. By extracting these features in different ways and using them in convolutional neural networks and decision trees we try to find out what characteristic features define different emotions. Besides this goal, we also investigate how the different artificial techniques can be used on this data, and from the differences that can be found we draw conclusions regarding the predictive value of energy, MFCCs and melfrequency spectra for different emotions. We conclude that more energy in the frequency range of 2400Hz to 4500Hz reliably correlates with the expression of happiness as opposed to sadness.

Contents

1 Introduction

Computer aided audio processing has been a hot topic of the last years and its applications can be found everywhere in the digital world today. Typical well-known applications might be Apple's 'siri' and Google Voice but also the speech-to-text in your car uses these techniques. Emotion detection, a subset of this research area, is used to detect the different emotions in speech which can be done reliably. However, while this does give us the correct emotions, this does not tell us why this emotion can be found in speech. In this bachelor thesis, we want to discover what features of speech expresses what emotion.

1.1 Current situation

Even though there are few real world examples of emotion recognition in speech to be found, emotion recognition is not something new, papers trying this can be found as early as 2003[KCHL03] with varying but overall good results. Recognizing emotion has thus been done before and has become pretty reliable too. Emotion recognition is generally done by some form of neural network, let it be a regular neural network, a convolution neural network or some form of LSTM. The last two of these perform generally the best, but these forms of neural networks are very good at only one thing, which is emotion recognition. What neural networks are poor at is describing why certain samples are classified as certain emotions. You could look at the weights of a neural network but overall it can be considered a black box.
If we want to start to have an insight in what features define emotions, there is a need for something we can understand by looking at it, thus not being a black box but a more transparent tool. The downside of this is that our currently known more 'transparent' techniques might not be applicable for recognizing emotion up to a certain standard anymore. What should be noted is that we already have some information on what defines a certain emotion in audio, this might not be clear from looking at the black box algorithms we use, but we can deduce this based on the input we provide. It is known that overall loudness is a pretty poor indicator of specific emotions[KJB+19]. Humans understand that someone talking loudly in a quiet environment might be more energetic and thus either happy or angry as opposed to for example sad. However overall loudness in audio files gets normalized and it is difficult for a microphone to hear that the environment is quiet while the person speaking is speaking loudly. Other aspects that are good indicators have also been found, such as melfrequency cepstrum coefficients (MFCCs), which will be discussed later on.

1.2 Thesis overview

This theses does not focus on classifying emotion, but rather on finding features that cause an piece of audio to sound like a certain emotion. This does not mean that we can skip the classifying part, we need be able to categorize emotion before being able to find out what makes them different. The goal of our thesis is thus to find which features have an effect on what emotion is heard by humans. To do this we propose the following plan:

- The first step is to gather fitting audio files.
- Then audio files will need to be appropriately converted to features.

- These text files will analysed with neural networks to show compatibility for emotion detection.
- Finally we analyse the features by constructing decision trees.

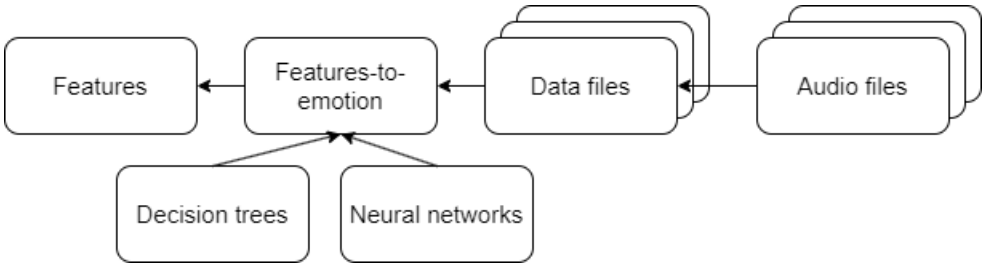

Figure 1: Proposed steps for extracting critical feature of emotions in speech.

The following sections will be divided as follows: Section 2 discusses related work and underlines the importance of audio recognition; Section 3 discusses the used audio files; Section 4 explains the used methods; Section 5 describes the experiments and different configurations; Section 6 will provide the results; Section 7 discusses the results and provides ideas for future research; Section 8 will conclude.

2 Related Work

In this section we will highlight previous work that can help us for our research. This section will be split into related work focused on emotion recognition and related work focused on feature extraction.

2.1 Emotion recognition

As mentioned above emotion recognition is an important part of this paper. This is not because we want to classify emotions, but if the computer can not classify the emotions this would mean that our techniques are also not able to point out what features determine emotion in speech, as the computer does not even know how to classify the emotions.

2.1.1 Neural networks

While their exist papers from as early as 2003, later papers show clear improvements such as The paper by Ingale et al. from 2012[IC12]. This improvement can be credited to either the new knowledge we have found for audio feature extraction in the meantime or the raw computing power that has become available over the years, allowing for more complex architectures and quicker neural network training. Ingale et al.[IC12] demonstrate different techniques for emotion recognition: Gaussian mixtures model, K-nearest neighbors, hidden markov model, support vector machine and an artificial neural network. The latest of these will be the main focus for us. Ingale et al. found that the gaussian mixtures model showed the best results, combined with support vector machines.

Artificial neural networks showed a poorer result with only 53% accuracy. Keep in mind that these were all tested on 6 different emotions. The true number of emotions will be discussed in the data set section.
The paper by Nicholson[NTN00] showed a rather large improvement to 50%, which is an improved result as they classified 8 emotions, although it is hard to compare accuracy percentages across multiple classification algorithms. So besides the number of emotions that are distinguished, the accuracies are hard to compare as different data sets are used and different neural network models which also might have been trained for different amounts of time. Even more speaker dependent versus speaker independent classification makes a difference too as speaker independent is generally more difficult, this difference will be discussed in the next paragraph. What we can conclude is that neural networks show to be at least a solid candidate for emotion recognition.

2.1.2 Convolutional neural networks and speaker dependent versus speaker independent

A recent paper (2019) by Zhao et al. [ZMC19] made use of both 1 dimensional and 2 dimensional (1D and 2D) CNN LSTM networks. These techniques will both be described in the methods and fundamentals section. This paper looked at 7 emotions and recorded accuracies of above 90% with their 2D network. Again, accuracies are hard to compare but 90% does show their methods do work, the lowest percentages reported were 89% for speaker dependent and 52% for speaker independent on a certain database. Speaker dependent here means that the database was first divided into different speakers and multiple networks were trained for these speakers, speaker independent is the opposite. Speaker independent results in lower accuracy scores as it is harder to classify emotion in speech for all people than for one person. A possible reason for this could be that people express their emotions in different ways. It is therefore easier to learn how a single person expresses their emotion than how people in general express emotions. For humans it is not too difficult to understand someones emotion, but a new person you have not met before might be a bit harder to guess. For the neural networks this seems to be similar.
A downside of convolutional neural network with LSTM besides the black-box issue discussed earlier is that design of different neural networks in terms of layers and activation functions can result in big differences in performance. However, this combination of neural network techniques did overall show promising results and is therefore a good option to show compatibility between our data set and emotion recognition.

2.2 Feature extraction

If we want to find characteristics of emotion in speech it is preferable to focus only on certain features of speech to decrease the complexity of a problem. What we can probably assume is that emotion can not be found with only a few features as input for the classifier. There is thus a trade-off between decreasing the removing features and increasing the performance of the emotion recognition. However, being able to decrease the feature set is more important for this paper as we try to find a general feature set of a certain emotion, having a perfect but complex classifier would mean that it is difficult to define this feature set. This only works up to a certain point and a somewhat reliable classifier is still needed.
Huang et al.[HDMZ14] created a different paper where they used a CNN but combined this with a

technique that is more fitting for our research. Their approach was made up of two stages. First candidate features were sought by a contrastive convolutional neural network, leaving them with only features that were deemed by the network to be useful as input features for emotion recognition. Then these selected features were used as input for another convolutional neural network where these selected features were used for the emotion recognition itself. The notable part here is the first step where Huang et al. tried to eliminate unnecessary features to decrease complexity. In our research where we are looking into what features make audio sound like a certain emotion this step is useful as it narrows our search. The downside here is network complexity as two separate networks need to be constructed and analysed. Furthermore, to be able to fully exclude features you need lots of data.
In the paper by Iliou et al.[IA09] their feature vectors were first subjected to a statistical analysis. This analysis allowed them to decrease their feature vector from 133 features to 35. In the second stage Iliou et al. made again use of a regular neural network, not a convolutional neural network, and the reported accuracy was %51. Whether the somewhat low accuracy is due to to their choice of neural network or removing of their features is unclear.

3 Database

In this section we will discuss the needed qualifications for the database and explain the used database was chosen.

3.1 Data set requirements

The base requirements that our data set should have are the following three:

- Each emotion must have enough samples for neural network training.
- The samples can not be acted.
- The samples have to be labeled by humans.

Thus besides having enough samples in general for neural network training we also need this to be balanced. If the data set is too unbalanced we will not have enough training samples for each of the emotions and than we can not gather results on every emotion. Having voice actors speak in certain emotions might give different results than having samples of real life emotional speech, which would lead us to discovering features of acted speech as opposed to regular speech. Finally, the samples have to be labeled by humans and not by another computer to have the exact interpretations of emotions that humans also have.

3.2 Emotions in the data set

The number of emotions that the data set differentiates is something to consider and not something with a clear right answer as this topic also tends to be a point of discussion for psychologists. In the last century, Paul Ekman published that there were 6 basic emotions: anger, disgust, contempt, happiness, sadness and surprise. However, there are also arguments for 27 basic emotions, even more, there are also arguments that there exist infinite emotions as emotion is a spectrum.

According to the paper by Dubois in 2015[DA15] which takes a neuropsychological perspective, emotion should be considered a spectrum, although the emotions can be categorized in 6 to 8 basic emotions. What was found lacking after classifying emotions in these basic emotions is the representation of dominance, arousal and valance. Additionally, there are more negative emotions than positive emotions in the 6 basic emotions, where happy and contempt are positive and the rest could be seen as negative emotions, although surprise could be considered both. Besides, researches are also debating on what emotions should are in the basic 6, for example whether fear is a basic emotion. In addition, it is also not entirely clear whether there are 6, 7 or 8 emotions. Similarly, not every bit of speech has a strong emotion, some are simply neutral. As it is so difficult to categorize emotions it would therefore be preferable if other aspects of the emotion was also noted, for example the valance or arousal. Having these extra labels would not necessarily improve our research, but this would show a certain quality of our data set as it does not to categorize all samples by 6 emotions, while this might not be complete.

3.3 Chosen databases

The database we ended up using is the MSP-Podcast corpus by R. Lotfian and C. Busso[LB19]. This data set is created by taking samples from podcasts, and labeling them by hand. By not using actors for the instances, this does mean that the emotion the speakers are expressing is not acted, but truly their emotion at that time. Furthermore, to acknowledge the paragraph above, the labeling in this podcast database includes valance, dominance and arousal, as well as fear and neutral emotions. There are 62,140 instances in this database. While this database is also labeled with the corresponding speaker, there are not an equal amount of instances per speaker nor does every speaker speak in every emotion. Therefore we can not do speaker dependent analysis but we can do speaker independent analysis. The data set included 10 different emotions: angry, sad, happy, surprise, fear, disgust, contempt, neutral, other and unknown. Other and unknown are however not very useful and will therefore be excluded during neural network training.

4 Methods and fundamentals

The methods and fundamentals section is split up into two sections. The first section explains how the database consisting of .wav files is converted to a database of .csv files with appropriate features. The second section discusses decision trees and introduces the usage of neural networks.

4.1 Audio conversion

As mentioned above, the database consists of over 62,000 instances. For these audio files to become suitable for both neural networks and decision trees we need to do some preprocessing. In addition, we also have to decide on what features to extract.

4.1.1 Ways of extracting audio

An important difference between decision trees and (convolutional) neural networks is that decision trees deal poorer with 2D data. Audio is in essence 2D as it is has a time dimension and feature dimension. We could flatten the data to be 1 dimensional however this would result in a very large

feature set on which it would be hard to perform analysis. We can also do additional preprocessing to achieve the same 1 dimensional feature set, which is discussed later.
Another issue is that the instances also have varying lengths, which is to be expected as the sentences in the podcasts also have varying lengths. Varying lengths is not necessarily an issue, but for time-dependent features this is an issue as they are not scaled to the same length, an example of this would be the sum of the energy in the sentence, a longer sentence can include more energy while not necessarily sounding louder. A remedy for this would be to split every audio file in parts of X milliseconds. Since we have varying audio lengths this would mean that our output vector will also have varying lengths, which is an issue for both the neural networks and decision trees. Therefore, we introduce two methods of dealing with the varying lengths.
The first option is that we cut the files into X equal parts, and concatenate the features that we extract, leaving us thus with a 1 dimensional feature set with the time-frames flattened, which can be used for decision trees.
The second method is to only use files of the same length. We decided for our paper to include only instances with a length of longer than 5 seconds and we cut those instances at 5 seconds. This left us with 37,339 files of 5 seconds long. Although it is not certain that the emotion can still be found in the cut sentence, we expect that 5 seconds should be enough to capture most of the emotion. Doing this left samples that can be seen in table 1.

Emotion	# of instances
Angry	1754
Sad	1393
Happy	7216
Surprise	1909
Fear	653
Disgust	1535
Contempt	2099
Neutral	13146
Total	**29705**

Table 1: Number of instances per emotion

With only files of exactly 5 seconds remaining we set our time-window length at X milliseconds to generate Y frames from each audio file. This thus results in a 2 dimensional data set, to input these in our decision trees we take the average over the time-windows. This input data is then 1 dimensional again. For the neural networks we do not apply this preprocessing. This data set will thus have all the time information in it, but when inputting it into the decision tree all the time information is removed.
In the end we will have three different data set formats:

- The 2 dimensional data set for the neural networks.
- The 1 dimensional data set with some time information for the decision trees.
- The 1 dimensional data set with no time information for the decision trees.

From here on forward we will refer to these as the full time-series, compressed time-series, and no time-series data sets. The no time-series and full time-series are thus essentially the same data set formats but averaged over the time-frames or not. For this reason the experiments for these will later be conducted together. Note that the full time-series and no time-series will also be made with two different sets of features, these are discussed in the next section.

4.1.2 Features to extract

As mentioned before, only selecting few features allows us to pinpoint better what features define a certain emotion. To extract the features we made use of opensmile [ope], a free open-source program that extracts features from audio by supplying it with a configuration file. Configuration files for emotion detection can already be found on their website, but these extract more features than wanted. Furthermore, for our first way of extracting features we have over 290,000 files after splitting them in 4 equal parts. Inputting this many files in opensmile would take very long if our configuration files are over complicated. The features that we ended up extracting were frequency spectra, MFCCs and energy. The exact configurations for these are discussed during the experiments.
Energy in a time-window in audio is equal to the magnitude of all frequencies combined in that time-window, this is comparable to the volume of this time-window. In our case we focus on the root-mean-square energy, or RMS energy. Which is calculated by taking the root of the average of the squares of the energy of the frequencies at that point in time. Earlier was mentioned that loudness is generally a poor indicator of an emotion in speech, however that does not mean it is the case for energy throughout time.
Frequency spectra, which are not the same as frequency cepstra, are spectra that show the intensity of the sound at a certain frequency at a certain point in time. Usually the frequency scale is not continuous but divided into small bin, as this allows for better use with computers. A human female voice reaches up to 17KHz and a male voice up to 8Khz[sea], however, reaching these levels is uncommon and people tend to speak more in the lower levels. Furthermore, people are better at distinguishing 1Khz from 2Khz than 7KHz from 8Khz. Therefore we change the scale of these bins from linear to either logarithmic or mel. This results in the logfrequency spectrum or melfrequency spectrum respectively. The general formula for mel is: $m = 2595log_{10}(1 + (f/700))$ where m is mel and f is the frequency in hertz. Both log and mel frequency spectra thus represent the same but their bins are separated at different intervals.

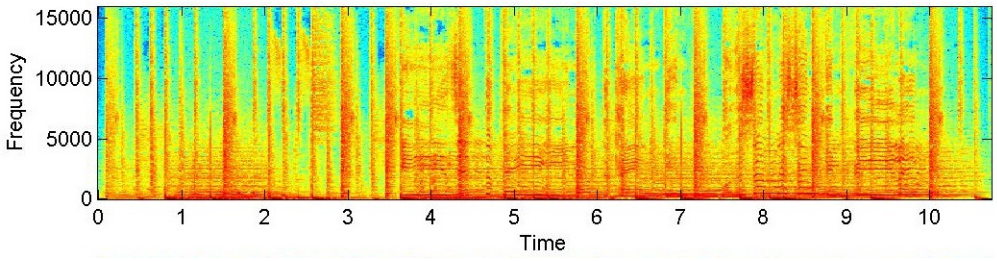

Figure 2: Frequency spectrum with a linear y-axis, source: [mod]

MFCCs, or melfrequency cepstrum coefficients are a more advanced feature of audio. MFCCs are the values that make up an MFC, or melfrequency cepstrum. A cepstrum, which are the first four letters of spectrum reversed, is a representation of the different energy levels spread over the different frequencies[Tiw10]. The mel part of melfrequency indicates that the frequencies are first mapped onto the mel scale, this scale is more comparable to how humans hear sound than the linear scale is. According to both the studies by Qawaqneh et al. and Molla [MH04][QMB17] MFCCs do not only mimic the human perception of sound very well, but are also effective for emotion recognition and speaker identification. As MFCCs have shown more promising than melfrequency spectra they are recorded, however since the regular melfrequency spectra are simpler to understand we also record these.

4.2 Decision trees and neural networks

Neural networks, as mentioned in the related works section, perform better in emotion classification than decision tree. Therefore we use neural networks to show that our data does contain information to base a reliable classifier on. Only if this can be done we can conclude that there is a relation between the data and the emotions. Once this has been done we construct decision trees to do the same classification task. It is expected that the decision trees perform worse than the neural networks. The reason for using the decision trees is that they are explicit about what features are good indicators for classification.

4.2.1 Neural networks

A neural network is a network consisting of sequential layers that each consists of nodes with a certain activation function. These nodes do a certain calculation and based on their input will give an output. The idea is by using multiple layers, which is a deep neural network, non-linear complex calculation can be made. The output of the second-last layer will be used as input for the final classification layer, which has a softmax activation function. This last layer will output probabilities corresponding to the likeliness that a sample is a certain emotion. For our research we create multiple neural networks as it is difficult to see what configuration for the neural network works best on the data set.
For our research our data can be in either one dimensional, with flattened time-frames, or two dimensional, not flattened. Both of these data types share the features axis, these features set contain the MFCCs and the melfrequency 'bins'. These features relate to each other in the sense that melfrequency bin 1 covers a lower frequency range the melfrequency bin 2, MFCC bin 1 covers a lower frequency range than MFCC bin 2. Note that energy will not relate to another feature in this sense. Moreover, in our two dimensional data not only do some of the features relate, but also the time-windows relate to each other, the first time-frame comes before the second time-frame etc.. These relationships between the features are similar to how pixels in a picture relate to each other. Therefore, we can apply a convolutional neural network, which has certain layers that do not calculate an output, but combine multiple of these features, either along the X, Y or both axis. This allows for the network to find patterns in the data that would not be found if the features were all treated as separate features.

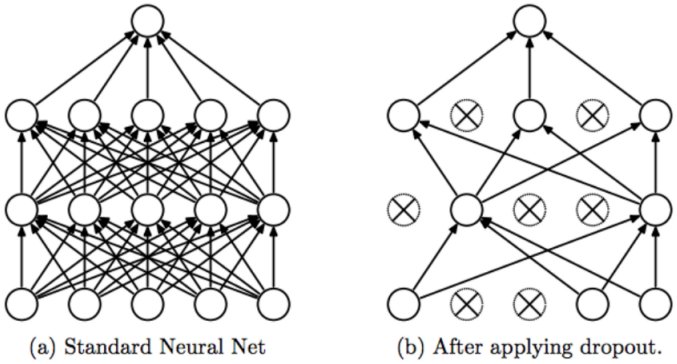

Figure 3: A neural network with dropout layers, source: [kdn]

4.2.2 Decision trees

A decision tree is an artificial intelligence technique where a tree is build with nodes which sort the input as good as possible. The bottom nodes, or leaves, in a decision tree do not split the data any further. All nodes show the most common class that can be found and certain properties such as the gini-index (impurity measurement) and instances at that node. The main advantage of decision trees for our research is the ease of interpreting the tree as each of the nodes in the tree will sort on a feature in our data set. The feature that is chosen by the tree to filter the data on tells us that this feature is the best way to split the data. In other words, that feature is the most important factor that differentiates two sets. Like for the neural networks, we create multiple decision trees for the problem as we do not know what works best for our data. The main parameter when creating decision trees is the max depth, which tells the maximum amount of nodes that can be chained below each other. Finally, with these decision trees we know what features separate certain emotion classes the best. With that information we can make conclusions to what features of speech are related to what emotions.

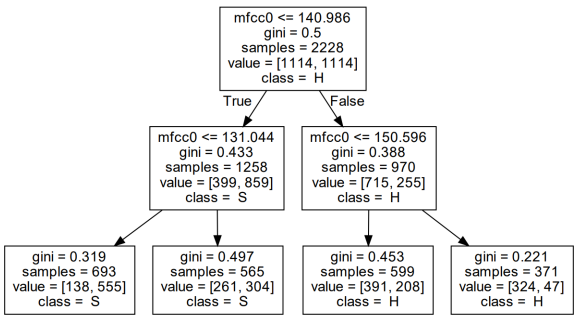

Figure 4: Tree with a max depth of 2, the root node is not counted

5 Experiments

In this section we will discuss the experiments that we have done.

5.1 Extracting features from audio

The first step for our experiments is extracting the features from the audio files itself. This can be split up into two parts. In the first part we show how we create the compressed time-series data, in the second part we show how the no time-series and full time-series data is created.

5.1.1 Splitting the audio in equal parts, compressed time-series data

As described in the methods section our data can be preprocessed in different ways with different selected features. For the compressed time-series data, where the data is divided into X pieces of equal lengths we decided to use 4 equal parts. Before filtering any data there where 73,043 files in the database. By dividing these into 4 equal parts we were left with 292,172 files with a total size of 12.2 GB, equal to the size of the complete 73,043 files.
The next step to extracting the features was creating a fitting configuration file for opensmile, the open-source audio feature extracting software mentioned above. For this configuration only melfrequency bins and energy where chosen to be extracted. For the Energy both the RMS and the Log are calculated, for the spectogram 8 bins are used which where divided along the Mel scale, starting from 20Hz and reaching up to 8Khz. Converting these audio from .wav files to .csv files decreased their size from 12.2GB to 69.2 MB. These .csv files were finally combined into one large .csv file with the emotion labels attached.
The configurations for opensmile and the scripts used can be found in appendix A.

5.1.2 Splitting the audio in frames, full time-series & no time-series data

As mentioned above for splitting the audio in frames of a certain small length we decided to only keep the 5 second long files. This resulted in the remaining 29,705 mentioned earlier, with other and unknown included this was 37,339 files. A new configuration file was used for this data set. The largest difference is that the framelengths are set at 0.050 secodonds which means that inputting 5 second long files will generate 100 frames. Following, the frames have been converted to gaussian shaped windows, this idea comes from the opensmile emotion recognition configuration files and allows for cleaner analysis of the frames as the time-windows do not cut off as abruptly. Furthermore, Energy_Log is no longer recorded as it describes the same feature as Energy_RMS, but in another format. Finally, we still only have 8 bins, if we want these bins to be as specific as possible we have to make the entire range as small as possible, therefore the frequency spectrum has been changed to the logarithmic scale and the high frequency now caps off at 6KHz. However, it is important to capture the entire frequency spectrum too, so this is a trade-off but we believe that the frequencies above 6KHz are rare and have little influence so more specific bins are more valuable.
The previous configuration files have described a melfrequency spectrum and a logfrequency spectrum. For the MFCCs another configuration file was created. This configuration used mostly the same settings as the one described above. However, the melfrequency bins have been increased to 26 and the calculation of 15 MFCCs has been added.

Finally, these methods result in 2D arrays. Based on the length of the frames we get a Y-axis with a length of 100 frames, and a X-axis with a length of 8 to 20 features, depending on if we extract melfrequency bins or MFCCs. Consequently, these will not be directly used for the decision trees. For the neural networks these files also need to be concatenated, preferably in an array of form (instances x timeframes x features). Therefore, each of the created files is read in and combined. This resulted in 2 files, for the melfrequency spectograms and MFCCs configurations, with sizes 338 MB and 625 MB respectively which is the full time-series data. To use these files with the decision trees the values were averaged over the timeframes column, this thus resulted in arrays of form (instances x averaged features) with 1 dimensional data per instance which is the no time-series data.
The configurations for opensmile and the scripts used can be found in appendix A.

5.2 Neural networks

We create two different sets of neural networks, one set for the MFCCs and one set for the melfrequency spectograms. The two neural network sets share the the same configuration for the neural networks and they both include 1D and 2D convolutional neural networks. The 1D convolutional neural networks use convolution only in the time-frame axis and does not find patterns between the features. The 2D convolutional neural networks use convolution in both axis and do try to find patterns between the features. Each of the neural networks is a sequential model and all models are constructed with keras, a machine learning library for python. The final layer in each of the models is a softmax layer, which calculates probabilities of each class, therefore the output size of it is also equal to the number of classes. In addition, each neural network uses dropout layers to prevent overfitting. Finally, besides convolutional layers also maxpooling is used to reduce the size of both the axis in 2D convolution, but mainly the time-frame axis as that is up to 100 time-frames long. In 1D convolution the maxpooling is only applied to the time-frame axis.
The downside for two dimensional convolution is that it is hard to introduce energy into the mix. Energy measures something different than the other features that relate to each other. Inputting it at the same level as the other features makes it treated as if it where the same kind of feature. In a maxpooling layer for example this might have as a result that if the energy feature is in a different order of magnitude it could always either overrule the other features, if it was much larger, or be neglected if it was much smaller. Using a different structure for our neural networks could have mitigated this issue but this was not tried.
Training is done with the Adam optimizer and the default learning rate is set to 0.01. A higher learning rate is favourable as it allows for quicker learning. However, having a too high learning_rate will impact the chance that our convolutional neural network correctly trains and converges to a maximum. The number of epochs is set to 200, more epochs will allow for better and more complete training but the point of diminishing returns is met before the 200 already. Besides, a perfectly classifying neural network is not our goal here, we want to prove that our extracted data allows for reliable classification. Finally, all of the neural networks were trained on the 2 dimensional data sets, which included the time-frames.
The configurations for the neural networks and the scripts used can be found in appendix B.

5.3 Decision trees

The trees in our research are meant to show which features and corresponding values show different emotions. It is not necessarily to be a good classifier although a higher accuracy does mean that the splits it has created are a good way of showing what features determine what emotion. The decision trees in our research were created with both data sets. With compressed time-series data to show relationships between features and emotions while using a simple form of feature extraction. The no time-series data was used to create decision trees that use the same data as the neural networks, albeit averaged over the time-windows. Differences in the performance between these two methods is discussed later on.
During the experiments for the decision trees there are a few different configurations that were tried. The following settings were changed during the experiments:

- Normalizing the data.
- Including the energy or not.
- Increasing the max depth of the trees.

Two different scripts are used for the two different types of data. These can both be found in appendix C.

5.4 Experiment configurations

With the basic experiments explained, the remaining part is which experiment configurations are used. For both the neural networks and decision trees multiple experiments are done. We can not try every single possible configuration because it would take far too much time. Instead for each of the experiments, the decision trees and neural networks, we define a list of settings that we experiment with in order from top to bottom. Once we have toggled through each of the possible options for a setting we continue with the setting that returned the highest accuracy score. By taking this greedy approach it could be that we miss a permutation of settings that resulted in even better scores. This is something we have to accept as a result of our experimentation approach.
Besides the possible settings for each of the experiments we also experimented with including all emotions or only selecting two. Including all emotions results in the best classifiers that can separate every emotion, the drawback is that this will result in lower accuracy scores as the classifiers have more classes to be confused by. Including only two emotion does thus not cover the entire possible spectrum of emotions, but does result in more reliable classifiers and still show us what differences in features between speech fragments could lead to different emotions. Again, we can not try every setting with every combination of emotions, therefore we first run the decision tree experiment with the 1 dimensional data to find out which pair of emotions is the easiest to separate for the classifiers. Not all the combinations for the different emotions are used when searching for the best separable pair, as with 8 emotions we would have 56 possible pairs. Instead every emotion together with happy is tried, The reason for choosing happy is that besides neutral it offers the most instances, and neutral might be a bit more vague being the 'center' emotion. The pair that is found is then used with the best performing neural network or decision tree while testing for all emotions.
Finally, for every possible experiment we always balance the data set as both the neural networks and decision trees can have issues with unbalanced data sets.

5.4.1 Compressed time-series data decision trees

The possible different settings for the compressed time-series data combined with the decision trees are:

- Combining the 4 sets of features of the 4 audio files into one set of features.
- Including the log energy, the RMS energy or no energy at all.
- Normalizing all the features to be range from 0-1, without energy normalized as well.
- Tree depth, ranging from 1 up to 5.
- Different sets of emotions.

5.4.2 Full & no time-series data experiments

In turn, the full and no time-series data sets are used for the neural networks and decision trees respectively. The different data sets, melfrequency spectra and MFCCs, each result in different experiments, therefore we run the entire set of experiments twice, once for each data set. The following possible sets of configurations is tried for the neural networks:

- Normalizing the data or not.
- Including the RMS energy or not.
- Using the two dimensional convolutional neural network.
- Different sets of emotions.

For the decision trees combined with the no time-series data we use the following sets of possible settings:

- Normalizing the data or not.
- Including the RMS energy or not.
- Changing the max-depth from 1 up to 5.
- Different sets of emotions.

6 Results

The full results can be found in appendix D, in this section we only highlight the best performing configuration for each of the sets of the experiment. Performance is measured with the accuracy score after measuring

6.1 Compressed time-series data decision trees

The following results have been generated by the best configurations for the decision trees using the compressed time-series data. The decision trees are made for all emotions together and the emotion that best contrasts the happy emotion. The best performing tree for all emotions can be found in 25. For this tree the following settings were used:

- We do combine the 4 sets of features of the 4 audio files into one set of features.
- We include only the RMS energy.
- We do normalize the data.
- Tree depth is set at 3.

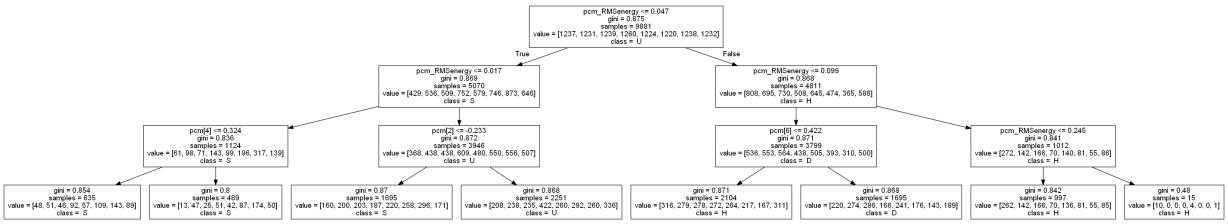

Figure 5: The decision tree for all emotions with an accuracy of 0.179

While creating the decision trees for the happy and a different emotion the same settings as above were used. The best distinguishable emotion pair was happy and sad. The corresponding decision tree can be found in 29

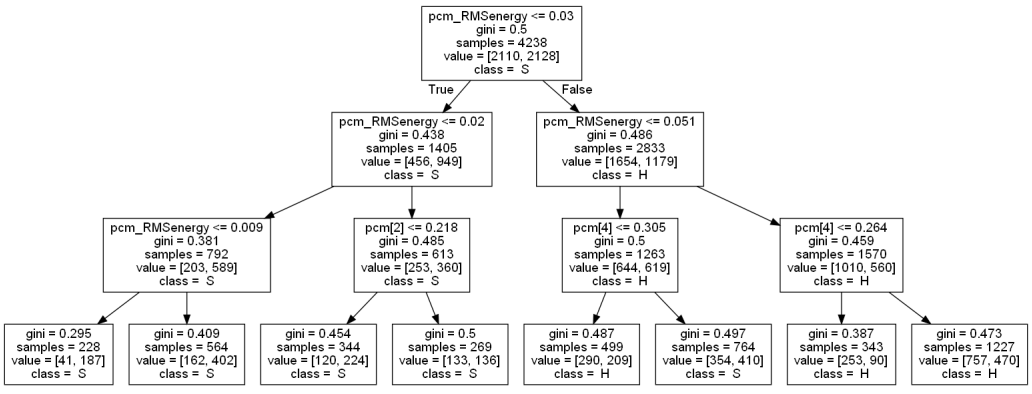

Figure 6: The decision tree for happy and sad with depth 3 and an accuracy of 0.624

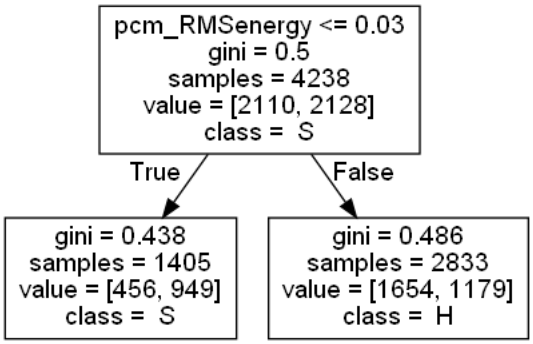

Figure 7: The decision tree for all happy and sad with depth 1 and an accuracy of 0.630

Multi class decision trees probably benefit from a different maximum depth than binary classification, therefore we also researched the best max depth for binary classification. The best performing tree can be found in 35. This used the same settings as the tree in 29 but the best performing max depth was found to be 1.

The reason that a tree might report a higher accuracy whilst using a lower max depth can be credited to the validation set, when constructing a tree with the training set it might implement a few splits that on the validation set do not work. Therefore simpler trees can sometimes be more generalizable and therefore report better accuracy scores.

6.2 Full time-series data neural networks

We have two different data sets that we use for the neural networks, the melfrequency data set and the MFCC data set. These data sets will be separately analysed for results as different configurations might work for different data sets. Each neural network will be trained both on all emotions, and on happy and sad alone as these achieved the highest accuracy in the compressed time-series data decision trees experiment. The data sets for the neural networks are thus the same data sets as for the no time-series decision trees, but then not averaged over the time-frames. These data sets are thus inputted as two dimensional data with all the time information present.

6.2.1 Melfrequency spectrum trained neural networks

The performance of the best performing neural network for all emotions can be found in figure 41. For that neural network the following configuration was used:

- We normalized the data set.
- We included the RMS energy.
- We used 1D convolution.

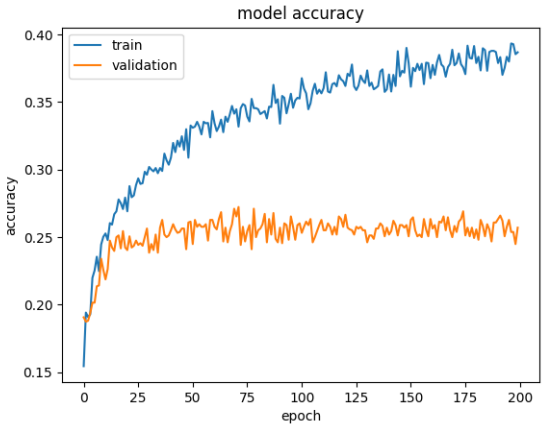

Figure 8: The neural network for all emotions with about a 25% maximum validation accuracy

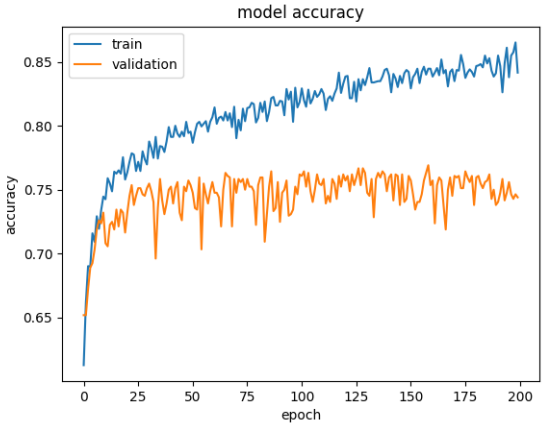

Figure 9: The 1D convolutional neural network for happy and sad with about a 75% maximum validation accuracy

We also tried these configurations for a neural network trained only on the happy and sad emotions, this resulted in the performance that can be found in 44.

6.2.2 MFCC spectrum trained neural networks

The performance of the best performing neural network for all emotions can be found in figure 46. For that neural network we used the following settings:

- We normalized the data set.
- We included the RMS energy.
- We used 1D convolution.

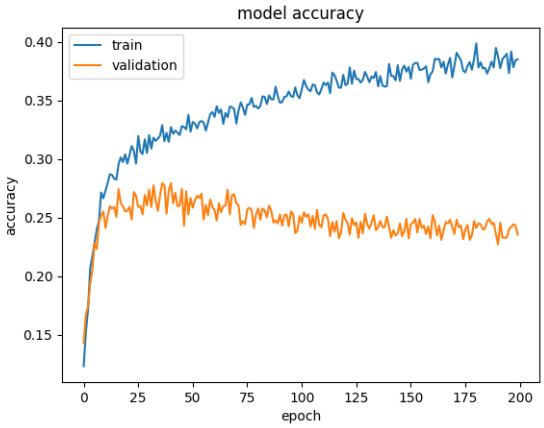

Figure 10: The 1D convolutional neural network for all emotions with about a 26% maximum validation accuracy after 200 epochs.

We also tried these configurations for only the happy and sad emotions which resulted in the performance that can be found in figure 49.

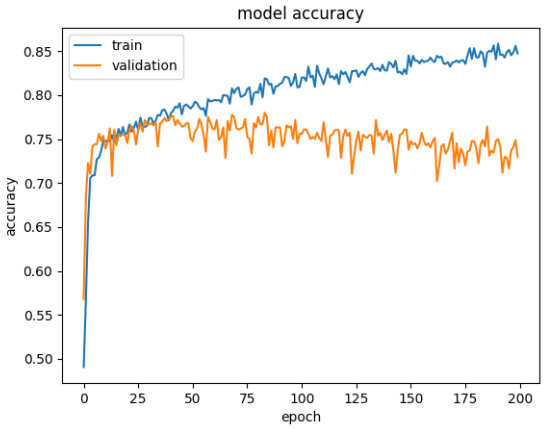

Figure 11: The 1D convolutional neural network for happy and sad with about a 75% maximum validation accuracy.

6.3 No time-series data decision trees

The final results to be presented regard the no time-series data used in decision trees. To use this data for the decision trees the values were averaged over the time-frames axis. Again, we have two sets of data and therefore two sets of results.

6.3.1 Melfrequency spectrum decision trees

The best performing tree for all emotions can be found in figure 60. For that decision tree the following settings were used:

- We did not include RMS energy.
- We set the max depth at 5.

This configuration was then also used for only happy and sad. This resulted in the tree that can be found in figure 61.

As this decision tree did not make use of the full max-depth of 5, we made another tree which can be found in 14. For generating this tree we removed the melfreq6 bin, this is done to see how important melfreq6 is compared to the other features when classifying. We also set the max-depth to 1 to ensure the same depth for both trees.

6.3.2 MFCC decision trees

For the MFCC data set results are created in the same way. The best performing tree can be found in figure 68. The following configuration was used for this decision tree:

- We did not include RMS energy.
- We set the max depth at 5.

This configuration was also used for the decision tree that only used happy and sad. The tree can be found in figure 69.
With all the results generated you can find the best performing accuracy score of each algorithm for all emotions and only for happy in sad in tables 2 and 3 respectively.

Algorithms	**Accuracy score**
Compressed time-series decision tree	# 17.9%
No time-series decision tree - Melfrequency spectrum	23.3%
No time-series decision tree - MFCCs	25.3%
Full time-series neural network - Melfrequency spectrum	25%
Full time-series neural notwork - MFCCs 26%	

Table 2: Best accuracy scores for each of the algorithms for all emotions

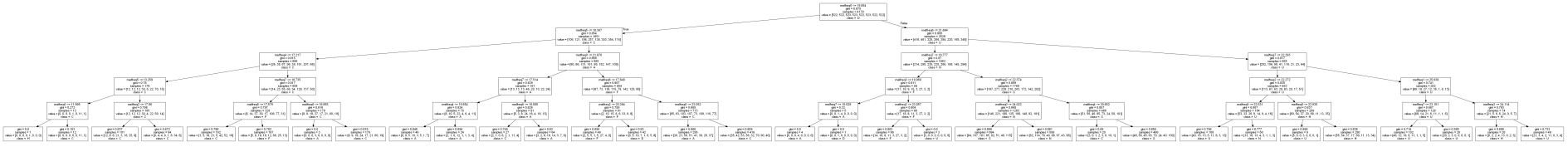

Figure 12: The decision tree for all emotions with an accuracy of 0.233

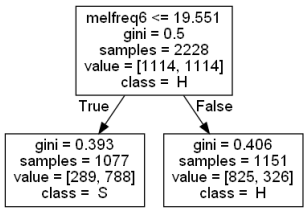

Figure 13: The decision tree for happy and sad with an accuracy of 0.728

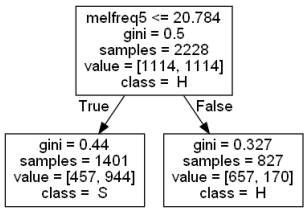

Figure 14: The decision tree for happy and sad with an accuracy of 0.686

7 Discussion and future Research

7.1 Compressed time-series decision trees

The decision trees that we have created based on the compressed time-series data sets for all emotions scored fairly low. Even when increasing the maximum depth tree to 5 the trees did not manage to score an accuracy above 18%. Randomly guessing classes would have resulted in a percentage of 12.5, therefore we can at least say that the data does show that we captured certain features of different emotions. However, remarkable is that in the first two layer of nodes, there are only splits based on the energy features if included. Energy is not necessarily a good indication of an emotion as it is bound to volume as well, which could be normalized differently by different microphones. Even more, a pattern in volume could be an indicator of emotion, but the time axis was not present in the decision trees and therefore such patterns can not be found. Besides this being the case, with enough samples this should average out and therefore we could somewhat reliably say that the energy is an indicator of the difference between happy and sad. This does align with the human interpretation of sad being a quieter emotion than happy.
The decision trees had the most trouble distinguishing happy and contempt or happy and surprise. This could insinuate that these emotions are more related than others, this could align with the human interpretations of these emotions: for example, happy and surprise both being somewhat active and possibly positive emotions. Happy and sad where the easiest to distinguish with the decision trees and this could also be true for people in general. Our tree managed to get an accuracy score of 62.4%, which is 12.4% above the random guessing rate. These results are not too promising either, but keep in mind that these trees were not meant as good classifiers but only to show which emotions were the easiest to distinguish, which as mentioned above we have found.
An issue for our research was that energy is a function that sums over time, therefore samples with a longer duration generally have more energy. Interesting is that the energy is still used for the splits in the top nodes. So, either there is a difference in volume between the emotions which can be still detected as the clips are on average the same length, or certain emotions tend to have longer

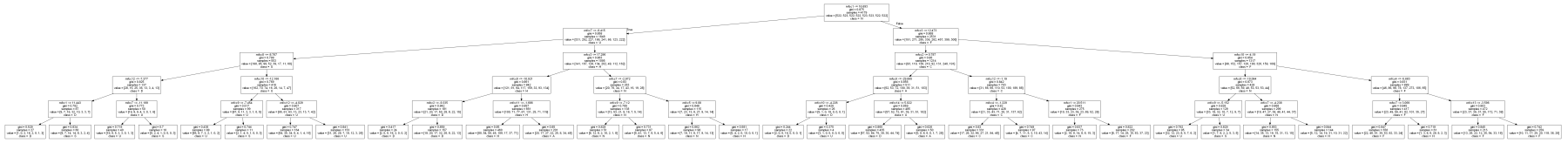

Figure 15: The decision tree for all emotions with an accuracy of 0.253

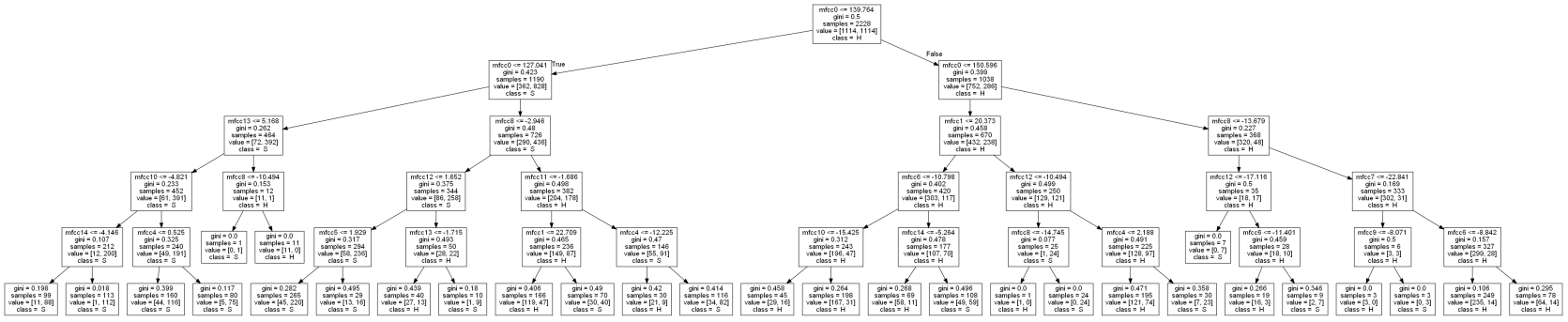

Figure 16: The decision tree for happy and sad with an accuracy of 0.747

Algorithms	**Accuracy score**
Compressed time-series decision tree	# 62.4%
No time-series decision tree - Melfrequency spectrum	72.8%
No time-series decision tree - MFCCs	74.7%
Full time-series neural network - Melfrequency spectrum	75%
Full time-series neural notwork - MFCCs	75%

Table 3: Best accuracy scores for each of the algorithms for only happy and sad

clips. This might mean that you need more time in a sentence for expressing a certain emotion, however, this is still very much guessing and the former seems more likely.
Finally, due to the limited time and computational power we could not analyse every emotion with every setting of the decision tree this would have been preferable so we could gather more data. Moreover, there might have been a certain configuration of decision trees and emotion that would show a higher accuracy than the 62.4% that we presented earlier. If this is the case than it would not necessarily be true that sad and happy are the easiest to distinguish, even more so as energy that is often used as the feature to split on is as mentioned above not necessarily very reliable. Again, the point of this part was only to somewhat reliable point out the best distinguishable pairs.

7.2 Neural networks

In this section we generated numerous neural networks to classify all the emotions as only happy and sad. Also the full time-series data was used which mitigated the energy issue mentioned above as all the frames have equal length, we now also incorporate all the time data. The different data sets, with either melfrequencies or MFCCs, performed fairly similar, the results were within 1% accuracy of each other. Using the energy feature of the data sets generally increased the accuracy by 1 or 2%. This is lower than expected as it seemed to be the most important feature for the decision trees created earlier. This effect can be credited to a few things, the first being that the convolution over the time-frames managed to extract patterns that a simple decision tree can not find. Another reason could be that the neural networks in general are better classifiers, which is to be expected, and that the energy is not necessarily needed anymore. Finally, energy is already incorporated in melfrequency and MFFC features so it is not actually new information.
Every neural network did seem to converge validation accuracy wise, but the training accuracy was not yet converted. This points to overfitting, this implies that it would have been possible to create simpler neural networks. Underfitting was not an issue as the best neural networks here classified with about 26% accuracy, which is 8 percent higher than the decision trees managed to do. This shows that the neural networks did find patterns in the data that decision trees could not find.
The use of 2D convolution showed very little use in our tests, the reason for this could be that our data is unfit for 2D convolution. Mixing features, energy and MFCCs for example can results in issues. RMS energy can be in a different order of magnitude than MFCCs, therefore during maxpooling this could have had extreme effects, either erasing the RMS energy or the nearby MFCC features. Furthermore, it does not make sense to mix features that are not related.
Finally, the reason for using the neural networks in our research was to show that there is actual ground for detecting emotions based on the features that we extracted. We believe that a 75% accuracy for binary classification shows that there is a connection to be found between our data

and the emotions, or at least for happy and sad. With 26% accuracy it is hard to call our classifiers reliable, however again we believe that there is more than enough reason to believe that there is a connection between our data and the emotions.

7.3 No time-series decision trees

Finally, we can determine what features make the difference between emotions. We could already do some of this with the 1 dimensional decision trees but these performed poorer and focused mainly on the energy which is not preferable. As mentioned, the no time-series decision trees showed better results than their compressed time-series counterparts and showed overall better results while not using the energy. That these trees performed better is a bit surprising, but this is most likely due to taking only samples with a length of 5 seconds. While the trees did not achieve the exact same accuracy as the neural networks, the accuracies were less than 1% lower. Although the neural networks could have been configured better, this still shows really promising results.
The MFCC data set did perform better than the melfrequency spectrum data, this difference can be credited to that MFCCs more closely relate to what humans hear than melfrequency spectra do. Furthermore, we collected almost twice as many MFCCs than melfrequency bins which thus does allow for more precise splits to be made. Most surprisingly is that including energy did not necessarily improve the decision trees' accuracy. This could have been a one-off error that would not have been present in a different set of configurations which we missed due to that we could not test everything. However, it at least does show that without energy we can also set up a, almost as reliable as a neural network, decision tree classifier. For the MFCC data set this could have been expected as MFCCs already incorporate energy information themselves, however for the melfrequency data set this is not the case.
Finally, the following knowledge has been gained from the results; Happy and sad are the easiest differentiable emotions, our data can be properly used for emotion classification and the generated decision trees show certain features to best split the classes on. By combining this knowledge we can safely look at the decision tree and draw conclusions from the features that it splits on. We will analyse the trees for the melfrequency spectrum and MFCC data sets separately. As the decision trees for all emotions only scored up to 25.3% the features that it splits on can not really be seen as good indicators of a certain emotion. However for the happy and sad decision trees we scored an accuracy of 74.7%. This is still incredibly reliable, but the splits will in our eyes at least show certain general features of an emotion.

7.3.1 Melfrequency spectrum features of emotions

For this analysis we refer to figure 61. First of all, this decision tree only made one split, even though the max depth was set at 5. This means that in the melfrequency data set the decision tree classifier could not find a sensible split besides splitting on melfreq6. The data set here splits from 1114 samples for both classes to two sets, (289, 788) and (825,326). After one split we end with two nodes, one with the sad emotion and one with the happy emotion. Validating this with our data set returned an accuracy of 72.8%, this is with a separate validation set of data, not the data on which the tree is built. When we generated another tree which can be found in 14 the accuracy dropped to 68.6%. While this is still a high accuracy, this does indicate that the melfreq6 bin is quite a bit more important when classifying then melfreq5 is. What is also interesting is that

melfreq5 is the frequency bin next to the melfreq6 bin, so it is not that a few random frequency ranges all give about the same results, but the best features to split on tend to be the ones with the higher frequencies.
The decision tree splits the samples on the melfreq6 feature with a value lower than 19.551. This melfreq6, or melfrequency6 feature corresponds with a certain frequency in the spectogram. We can find what frequency corresponds to what melfrequency bin by displaying the information during feature extraction with opensmile:

```
instance 'melspec': Band -1 (left bound) : center = 20.000000 Hz
instance 'melspec': Band 0 : center = 222.508819 Hz ; bw/2 = 230.987885 Hz
instance 'melspec': Band 1 : center = 481.975770 Hz ; bw/2 = 295.956065 Hz
instance 'melspec': Band 2 : center = 814.420950 Hz ; bw/2 = 379.197269 Hz
instance 'melspec': Band 3 : center = 1240.370308 Hz ; bw/2 = 485.851152 Hz
instance 'melspec': Band 4 : center = 1786.123253 Hz ; bw/2 = 622.502799 Hz
instance 'melspec': Band 5 : center = 2485.375906 Hz ; bw/2 = 797.589345 Hz
instance 'melspec': Band 6 : center = 3381.301942 Hz ; bw/2 = 1021.921129 Hz
instance 'melspec': Band 7 : center = 4529.218164 Hz ; bw/2 = 1309.349341 Hz
instance 'melspec': Band 8 (right bound) : center = 6000.000625 Hz
```

Figure 17: Frequency ranges of the melfrequency bins

The bands show their center frequency and their bandwidth on either side, therefore the bandwidth is shown as bandwidth/2. Band 6 ranges from frequency 2360 Hz up to 4402 Hz, the bands thus also overlap a little bit. From this we can conclude that in general happy emotional speech has more energy in the frequency spectrum between 2360 Hz and 4402 Hz than sad emotional speech does. We can not conclude how much more, as we can not read that from the figure. Furthermore we also do not know if that is what makes the speech sound happy or sad, but we do know that these frequencies can be heard more in happy emotional speech.

7.3.2 MFFC features of emotion

For this analysis we refer to figure 69. Due to compression issues the visibility of this figure is quite worse, we therefore include this picture separately as mfcctree.png. As opposed to the melfrequency spectrum decision tree, this tree did generate multiple splits. The order of these splits is important as the earlier features tend to be the best features to split the data set on. However, the later nodes can still present valuable information. For that reason we discuss the layers in order from top to bottom. We can judge the effectiveness of a certain split by looking at the gini-index, a lower gini-index represents a better split. However, when there are only a few samples remaining this generally does not indicate too much and the results are probably not generalizable, therefore we basically neglect all the nodes with less than 100 samples remaining.
Although it is possible to discover what MFCC feature relates to what frequency range, this is much more difficult than for the melfrequency bands. The advantage of using MFCCs was that the data extracted corresponds more with what humans hear. Furthermore, unlike melfrequency band values these values can both be positive and negative, where higher values in an MFCC mean that

there is more energy to be found in a the higher frequencies. In addition, higher values also mean that there is more energy to be found in a certain band.
The first split in our decision tree splits on mfcc0, which is our first MFCC band, as do the splits in the second layer. Interesting is that at each of the splits the sad samples are more split to the left and the happy samples are more split to the right. Therefore we can say that in general, happy emotional voices have a higher value regarding the first MFCC band. This property is true in almost every split and we can therefore also see that after each split the primary class on the right is often happy and the class on the left is often sad, although there are some deviations. Beyond the first splits there is no feature that is much more used than the others. This indicate that beyond mfcc0 there is not necessarily a best separating feature.
MFCC features represent not only the frequency, but also the energy. In the first decision trees we already concluded that happy emotional speech had higher energy levels than the sad emotional speech. With the happy samples generally ending on the right after each split we see this relation again, as each of these splits carry some of the energy information as well since higher values mean more energy. This is reinforced by the fact that the 6 left-most leaves all have as primary class sad and that the samples in these leaves probably have the least energy.

7.4 Future research and limitations

For future research it might be interesting to take sample length as a feature during classifying, this is of course a horrible classifier overall, but it could answer the question whether certain emotions take generally more time to express. Furthermore, the energy calculation should during further research also be changed to properly incorporate different frame lengths. In addition, a simpler neural network but with a separate input layer for the energy feature could enhance the performance of 2D convolution. To create better performing neural networks a good option would be to create more melfrequency bins so that the features are more fit for convolution. Another improvement could be that if some emotions are harder to differentiate than other emotions, it could be possible that certain emotions are more related. This information might be useful for neural network training as we could penalise classifying happy as sad harsher than classifying happy as contempt. Changing the focus of classifying emotions to classifying valance with regression could lead to the same advantages.
In future research if we were able to increase the accuracy of the methods for all emotions we could show not only what features make the difference between happy and sad but also could show at once what features make the difference between happy and all other emotions. Another way to do this would be to look at all the permutations, but for that we currently run into time or computational power issues.
Finally, by combining this information with the information found during the melfrequency decision trees we believe that this could be used as a starting point for future research for converting happy emotional speech to sad emotional speech. We believe that by changing the numerical data the emotion of speech can be changed. However, this is very much speculation, and patterns or a certain rhythm that might be in a certain emotion would also need to be changed, which is something that with our methods we have not been able to pick up on.

8 Conclusion

To conclude, emotion detection in speech can be done fairly reliable with only a few features measured. Binary classification reached up to 75% accuracy with only 15 features measured, and for 8 class classifying an accuracy score of 26% was reached. This shows that the features we extracted, whether this included energy or not, did provide a good basis for detecting emotion. We expect that by increasing the number of features and a better neural network structure these accuracies will only improve.

We have also discovered that MFCCs represent the emotions better than melfrequency bins, which was also mentioned in earlier related work and in turn resulted in better classifiers. However, the performance of melfrequency spectra did still result in a somewhat reliable classification. Energy levels in voice are a useful feature to extract but microphones could have an effect on these. Besides this, energy is already in more detail with frequency spectra. Therefore only when not using the other features is there a case for recording energy. Although some of our decision trees used only energy to classify emotion, we believe that there are more differences between happy and sad than the overall energy levels.

Regarding our research methods, convolutional neural networks provided an advantage over decision trees when it comes to classifying accuracy. However, the simple interpretation possibilities of decision trees do provide a good reason for using them in this setting. Besides that, our 1D convolutional neural networks for these data sets did seem to be overfitting, therefore a reduction in the complexities of the neural networks could be made. This should make the validation and test accuracies align more. 2D convolutional neural networks did not seem to provide any other advantage for our research, however we believe that this is due to the amount of features we have extracted and the configuration of the 2D convolutional neural networks. Hence, with improvements in these areas we expect 2D convolution to improve much more, we are not sure yet if it would surpass 1D convolution and for that more research is needed. For this paper we used 2 different data sets which where created by extracting different features from the database provided. The MFCC data set seemed to be the most promising in terms of classification.

Finally, we have found a few differences between features in emotions. Due to our experiment limitations we could not generate clear results for each of the emotions, but we have found differences in features between happy and sad. As the first MFFC band is the primary filter for the decision tree and the way the data set splits, we can conclude that happy emotional speech uses more energy in the lower frequency band when compared to sad emotional speech. In addition, as melfreq6 was the only filter in the melfrequency decision tree we can conclude that happy emotional speech uses more of the frequency spectrum between 2360 Hz up to 4402 Hz. When testing for other melfrequency bins as well it turned out that especially melfrequency bin 6 is a good indicator.

References

[DA15] Julien Dubois and Ralph Adolphs. Neuropsychology: how many emotions are there? *Current Biology*, 25(15):R669–R672, 2015.

[HDMZ14] Zhengwei Huang, Ming Dong, Qirong Mao, and Yongzhao Zhan. Speech emotion recognition using cnn. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 801–804, 2014.

[IA09] Theodoros Iliou and Christos-Nikolaos Anagnostopoulos. Statistical evaluation of speech features for emotion recognition. In *2009 Fourth International Conference on Digital Telecommunications*, pages 121–126. IEEE, 2009.

[IC12] Ashish B Ingale and DS Chaudhari. Speech emotion recognition. *International Journal of Soft Computing and Engineering (IJSCE)*, 2(1):235–238, 2012.

[KCHL03] Oh-Wook Kwon, Kwokleung Chan, Jiucang Hao, and Te-Won Lee. Emotion recognition by speech signals. In *Eighth European conference on speech communication and technology*, 2003.

[kdn] Don't use dropout in convolutional networks.

[KJB+19] Ruhul Amin Khalil, Edward Jones, Mohammad Inayatullah Babar, Tariqullah Jan, Mohammad Haseeb Zafar, and Thamer Alhussain. Speech emotion recognition using deep learning techniques: A review. *IEEE Access*, 7:117327–117345, 2019.

[LB19] R. Lotfian and C. Busso. Building naturalistic emotionally balanced speech corpus by retrieving emotional speech from existing podcast recordings. *IEEE Transactions on Affective Computing*, 10(4):471–483, October-December 2019.

[MH04] K.I. Molla and K. Hirose. On the effectiveness of mfccs and their statistical distribution properties in speaker identification. In *2004 IEEE Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2004. (VCIMS).*, pages 136–141, 2004.

[mod]

[NTN00] Joy Nicholson, Kazuhiko Takahashi, and Ryohei Nakatsu. Emotion recognition in speech using neural networks. *Neural computing & applications*, 9(4):290–296, 2000.

[ope] Opensmile.

[QMB17] Zakariya Qawaqneh, Arafat Abu Mallouh, and Buket D. Barkana. Deep neural network framework and transformed mfccs for speaker's age and gender classification. *Knowledge-Based Systems*, 115:5–14, 2017.

[sea] Human voice frequency range.

[Tiw10] Vibha Tiwari. Mfcc and its applications in speaker recognition. *International journal on emerging technologies*, 1(1):19–22, 2010.

[ZMC19] Jianfeng Zhao, Xia Mao, and Lijiang Chen. Speech emotion recognition using deep 1d 2d cnn lstm networks. *Biomedical Signal Processing and Control*, 47:312–323, 2019.

A Extracting features

For extracting features we made use of three different configuration files, these configuration files are 8bucketsmel.conf 8bucketscnn.conf and 8bucketscnnmfcc.conf. These correspond to the compressed time-series data set, the full time-series melfrequency spectrum data set and the full time-series MFFC data set respectively. The configuration files can be found separately included. We will here highlight the most important configuration settings.

Setting	**Value**
Framelength	# 50
Energy_RMS	True
Energy_log	True
Spectogrambins	8
Spectogramscale	Mel
Low frequency	20
High frequency	8000

Table 4: Configuration settings for compressed time-series data set

Setting	**Value**
Framelength	# 0.050
Windowfunction	# gauss
Sigma	# 0.25
Gain	# 1
Energy_RMS	True
Energy_log	False
Spectogrambins	8
Spectogramscale	Log
Low frequency	20
High frequency	6000

Table 5: Configuration settings for full time-series melfrequency spectrum data set

For the 1 dimensional data set we split the files into 4 equal parts, this was done with Audio_splitter.py, the files where later combined again with 1combine.py.
For all the data sets we input all the files into opensmile with 1opensmile_script.py. To only select the labels from the labels file in the database we used main_only.py. For combining the different output files of opensmile we used 2combined.py. To attach the labels we then finally used Combine_labels.py.

B Neural networks

The neural networks are defined in two different files. cnnmelfreq.py and cnnmfcc.py for the melfrequency spectrum data set and MFCC data set respectively. Both of these files use the same settings and will therefore be only explained once. These settings are:

Setting	Value
Framelength	# 0.050
Windowfunction	# gauss
Sigma	# 0.25
Gain	# 1
Energy_RMS	True
Energy_log	False
Spectogrambins	26
Spectogramscale	Log
Low frequency	20
High frequency	6000
first MFCC	0
last MFCC	14

Table 6: Configuration settings for full time-series MFCC data set

- run_cnn
- save
- balance
- normalize
- add_energy
- load_model
- twodconv

run_cnn determines whether a neural network gets build, toggling this off allows for quicker run time of the file. save determines whether a new neural network gets saved after training or not. balance determines if the data set gets balanced according to the emotions. add_energy determines if the energy feature is added during the training. load_model changes the behaviour from creating a neural network to loading a saved one in. twodconv toggles between 2D convolution or 1D convolution.

C Decision trees

The decision trees are also split over two files, Tree.py is used for the 1D decision trees and Treeformfcc.py for the 2D data sets. tree.py has the following settings:

- run_tree
- split
- log

- normalize

run_tree determines whether the tree is created and saved. split determines if there is created a seperate validation set. combine_audios combines the 4 different sets of features into 1 set of features by adding them together. log determines whether the log energy feature is included. normalize finally decides if the data gets normalized or not before setting up the tree.
Treeformfcc.py has the following settings:

- run_tree
- save
- allemotions

run_tree determines whether the tree is created. save determines whether the newly created tree is saved. allemotions is a list where you can input which emotions should be included. To determine if energy needs to be included we simply make a comment of line 56.

D Results

Here we will show every single result that we have generated, keep in mind that we have not tried every permutation for the settings as we selected the settings when generating the results with a greedy approach.

D.1 Compressed time-series data decision trees

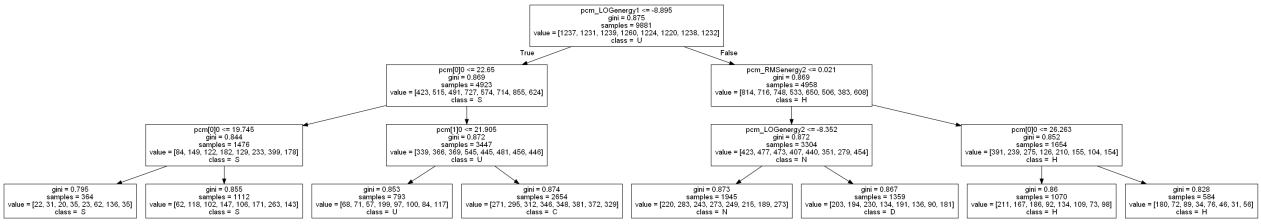

Figure 18: The decision tree for all emotions, with the 4 sets of features not combined, both energy features included, the data not normalized and the depth set at 3. This tree had an accuracy of 0.128

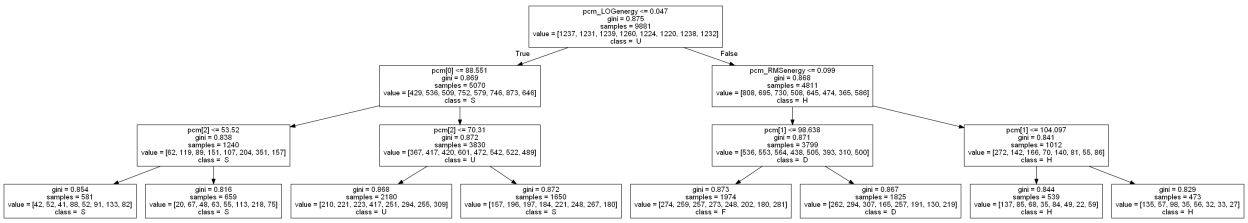

Figure 19: The decision tree for all emotions, with the 4 sets of features combined, both energy features included, the data not normalized and the depth set at 3. This tree had an accuracy of 0.176

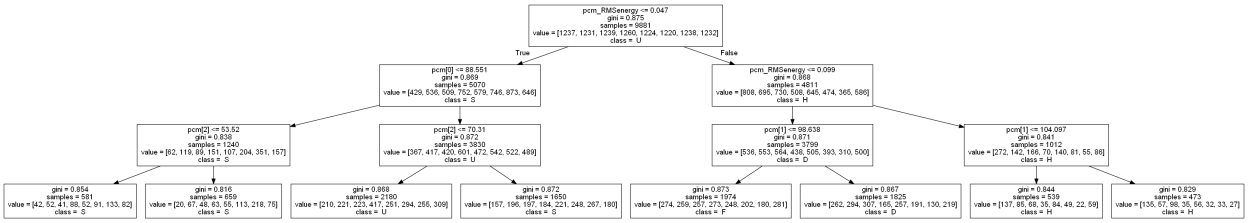

Figure 20: The decision tree for all emotions, with the 4 sets of features combined, only RMS energy included, the data not normalized and the depth set at 3. This tree had an accuracy of 0.176

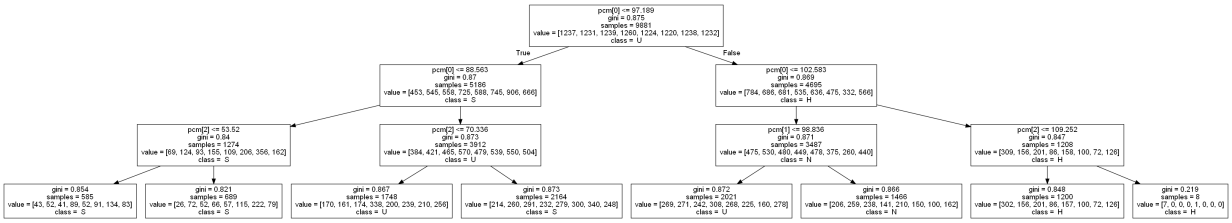

Figure 21: The decision tree for all emotions, with the 4 sets of features combined, no energy included, the data not normalized and the depth set at 3. This tree had an accuracy of 0.172

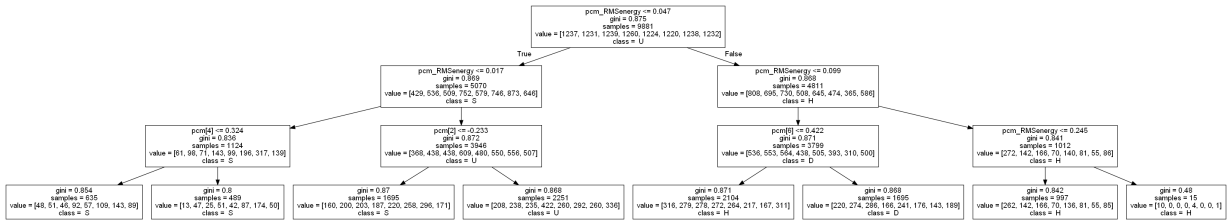

Figure 22: The decision tree for all emotions, with the 4 sets of features combined, RMS energy included, the data normalized and the depth set at 3. This tree had an accuracy of 0.179

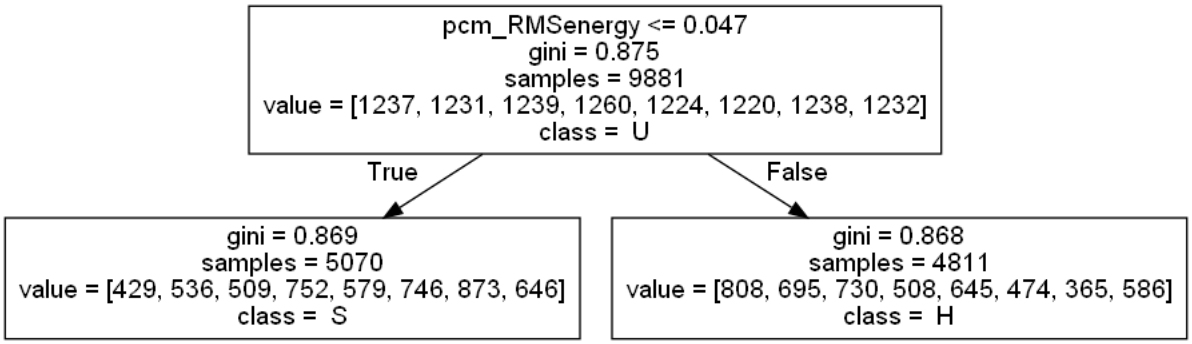

Figure 23: The decision tree for all emotions with depth 1 and an accuracy of 0.166

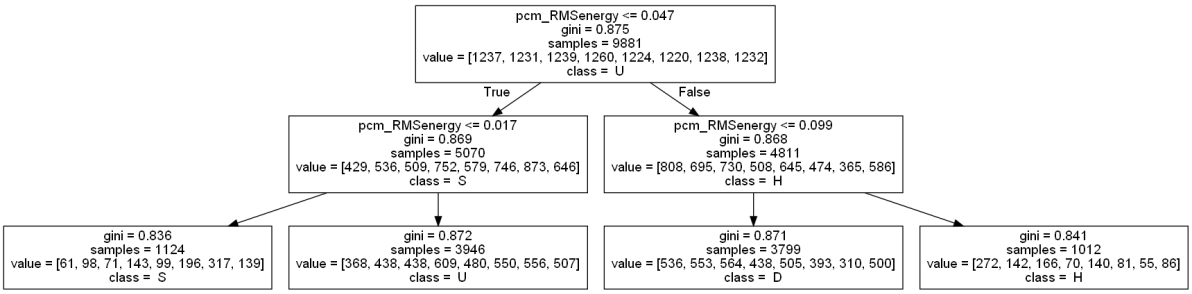

Figure 24: The decision tree for all emotions with depth 2 and an accuracy of 0.165

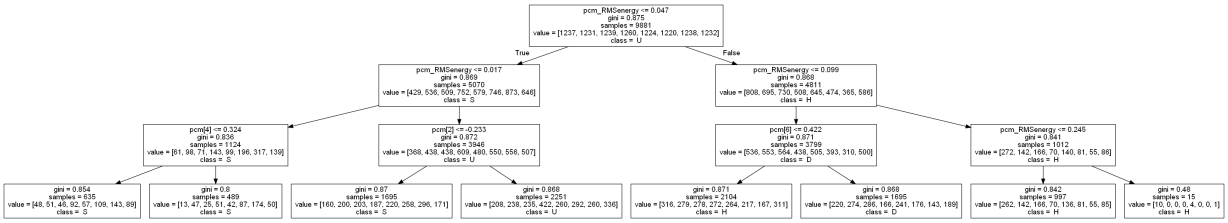

Figure 25: The decision tree for all emotions with depth 3 and an accuracy of 0.179

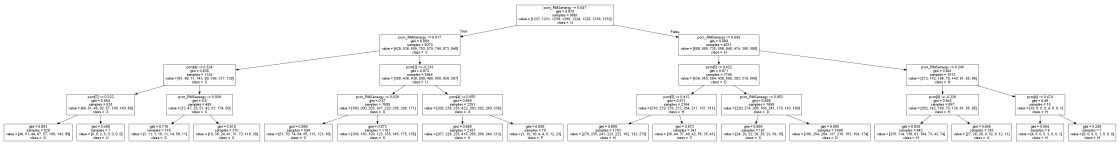

Figure 26: The decision tree for all emotions with depth 4 and an accuracy of 0.179

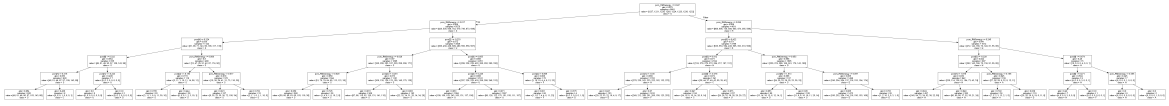

Figure 27: The decision tree for all emotions with depth 5 and an accuracy of 0.172

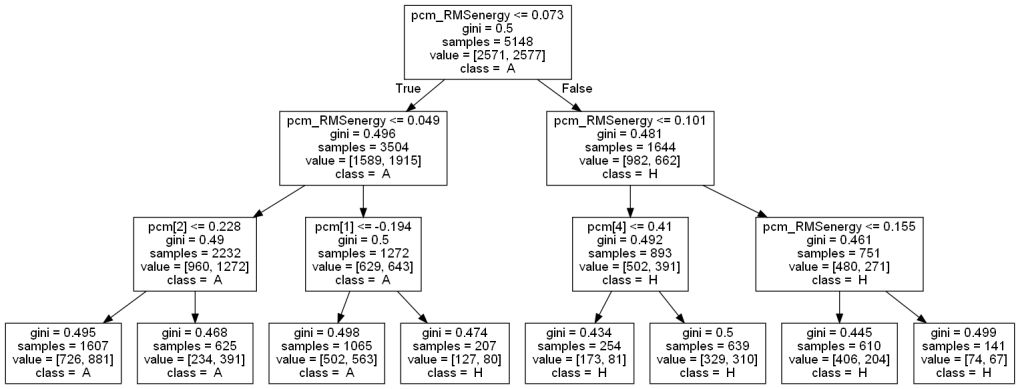

Figure 28: The decision tree for happy and angry with depth 3 and an accuracy of 0.563

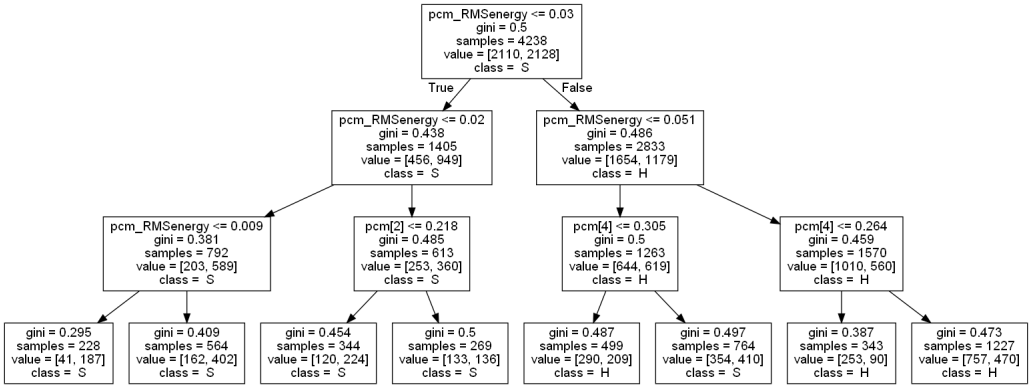

Figure 29: The decision tree for happy and sad with depth 3 and an accuracy of 0.624

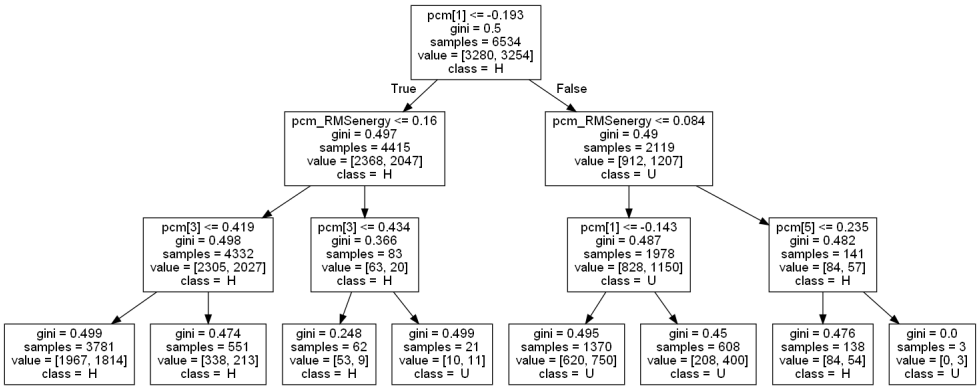

Figure 30: The decision tree for happy and surprise with depth 3 and an accuracy of 0.526

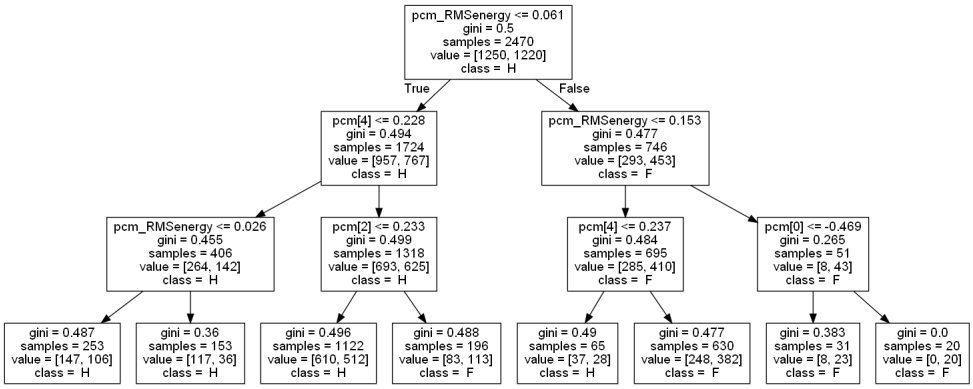

Figure 31: The decision tree for happy and fear with depth 3 and an accuracy of 0.578

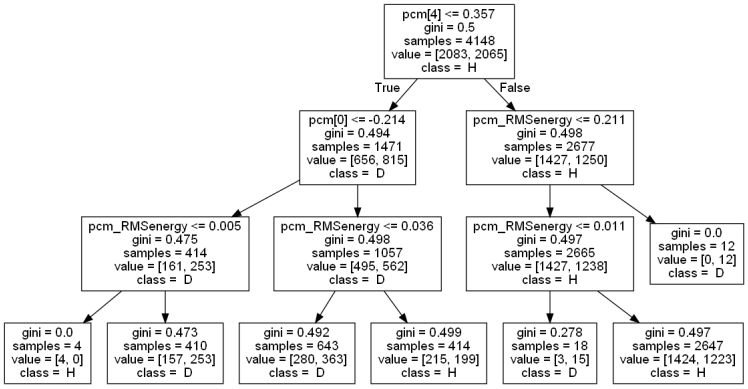

Figure 32: The decision tree for happy and disgust with depth 3 and an accuracy of 0.558

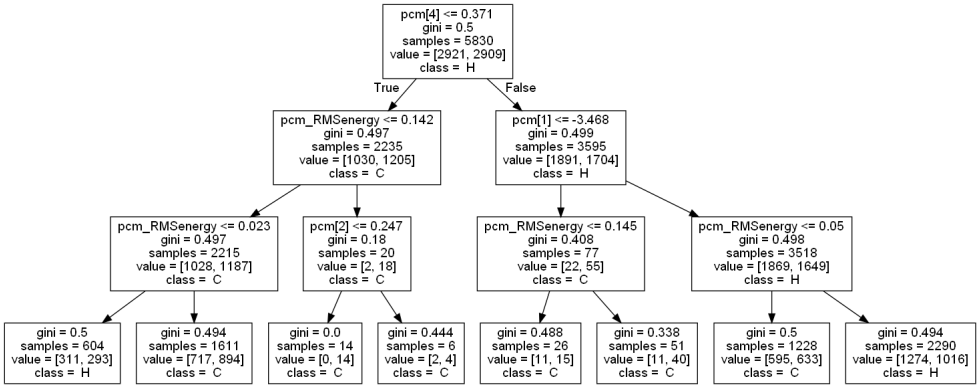

Figure 33: The decision tree for happy and contempt with depth 3 and an accuracy of 0.528

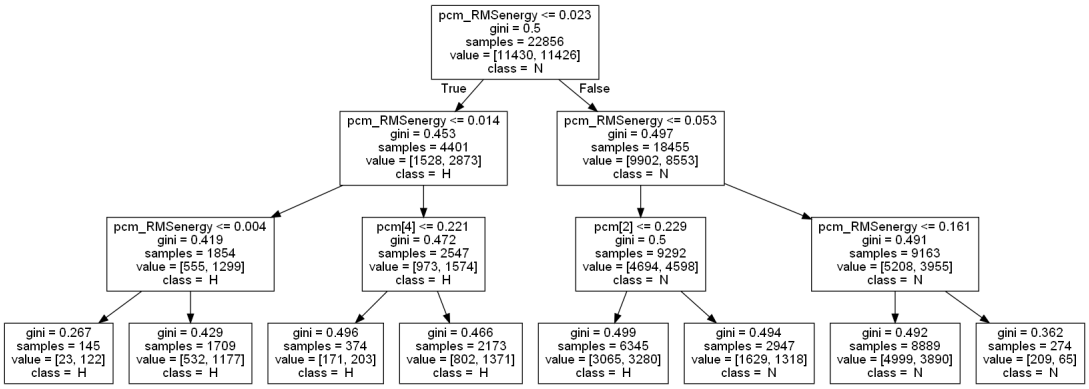

Figure 34: The decision tree for happy and neutral with depth 3 and an accuracy of 0.562

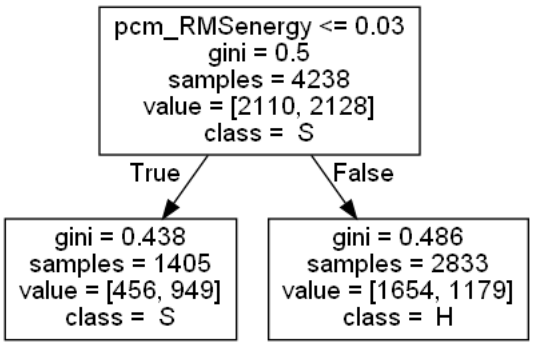

Figure 35: The decision tree for all happy and sad with depth 1 and an accuracy of 0.630

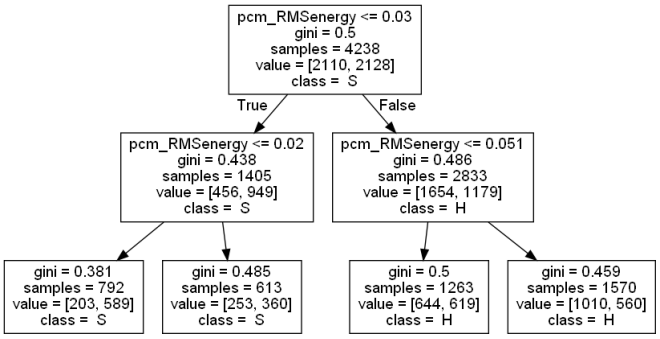

Figure 36: The decision tree for happy and sad with depth 2 and an accuracy of 0.630

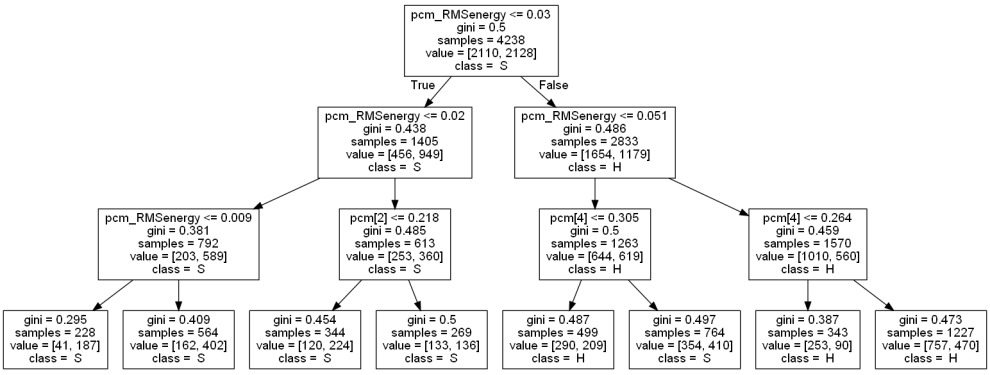

Figure 37: The decision tree for happy and sad with depth 3 and an accuracy of 0.624

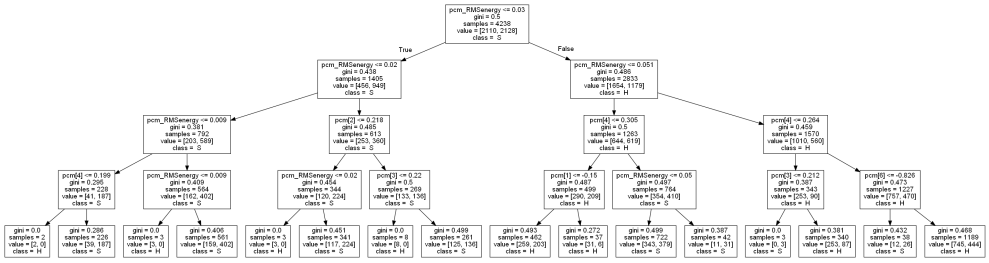

Figure 38: The decision tree for happy and sad with depth 4 and an accuracy of 0.625

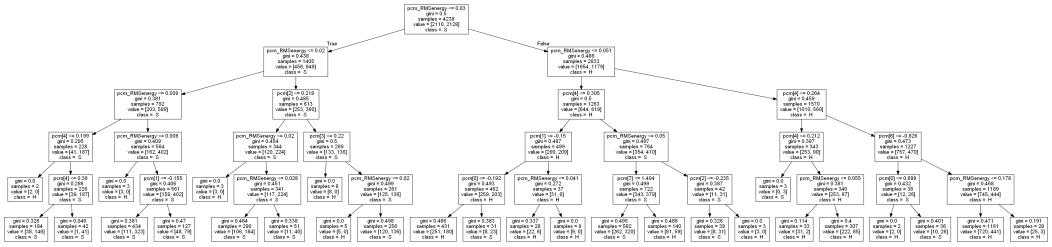

Figure 39: The decision tree for happy and sad with depth 5 and an accuracy of 0.625

D.2 Neural networks

D.2.1 Melfrequency neural networks

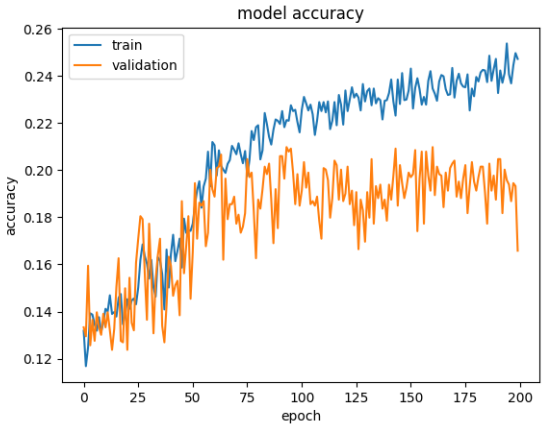

Figure 40: The 1D convolutional neural network for all emotions, with about a 20% maximum validation accuracy

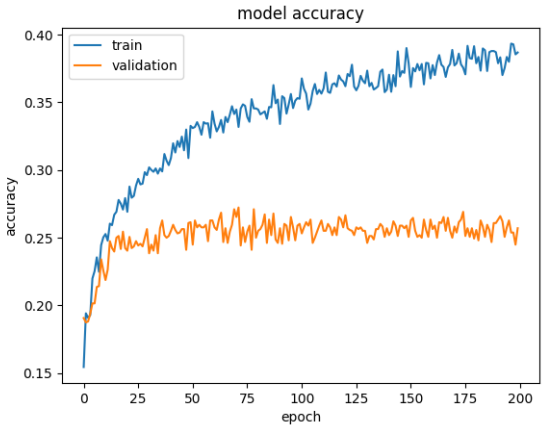

Figure 41: The 1D convolutional neural network for all emotions with normalization, with about a 25% maximum validation accuracy

D.2.2 MFCC neural networks

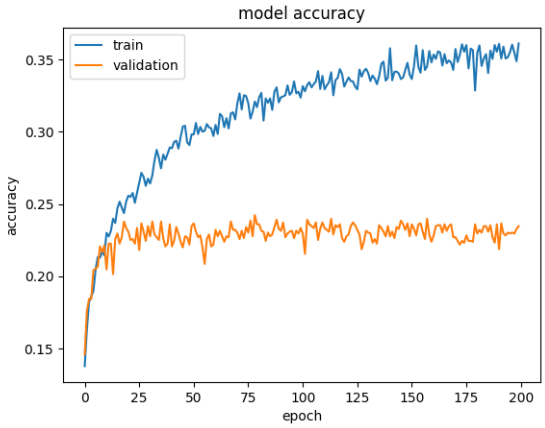

Figure 42: The 1D convolutional neural network for all emotions with normalization without energy, with about a 23% maximum validation accuracy

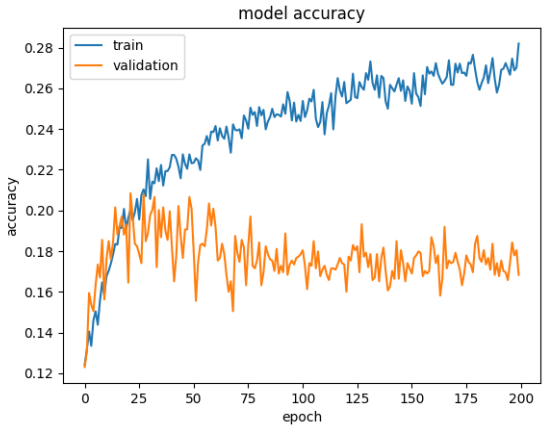

Figure 43: The 2D convolutional neural network for all emotions with normalization and energy, with about a 19% maximum validation accuracy

D.3 No time-series data decision trees

As the accuracies are nearly identical we chose to continue here with the tree that uses the least amount of features, so the tree from figure 55

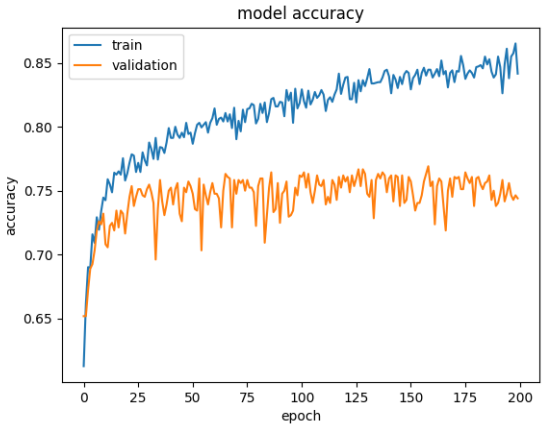

Figure 44: The 1D convolutional neural network for happy and sad with normalization and energy, with about a 75% maximum validation accuracy

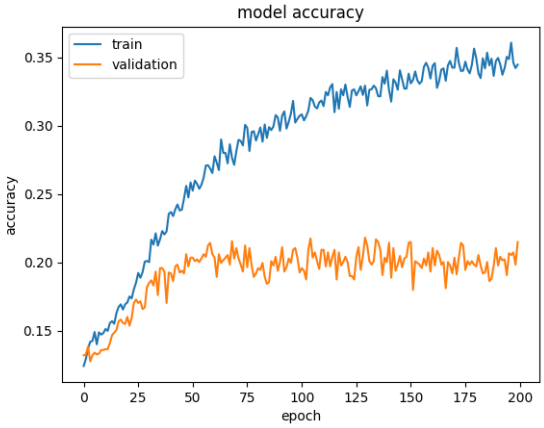

Figure 45: The 1D convolutional neural network for all emotions, with about a 21% maximum validation accuracy

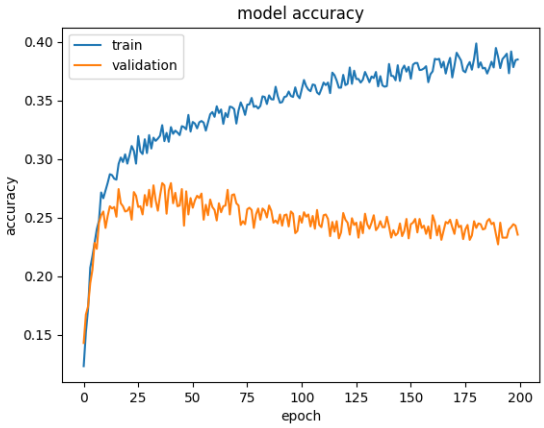

Figure 46: The 1D convolutional neural network for all emotions with normalization, with about a 26% maximum validation accuracy after 200 epochs.

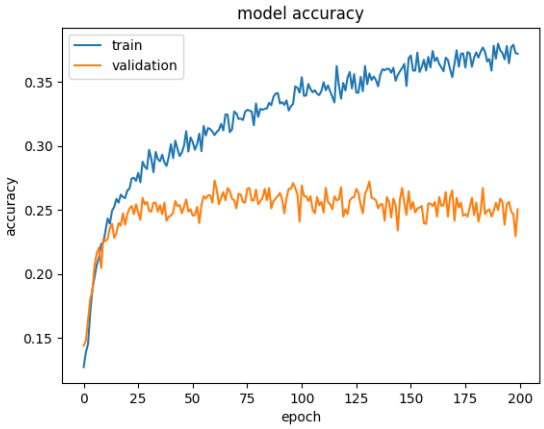

Figure 47: The 1D convolutional neural network for all emotions without energy with normalization, with about a 25% maximum validation accuracy.

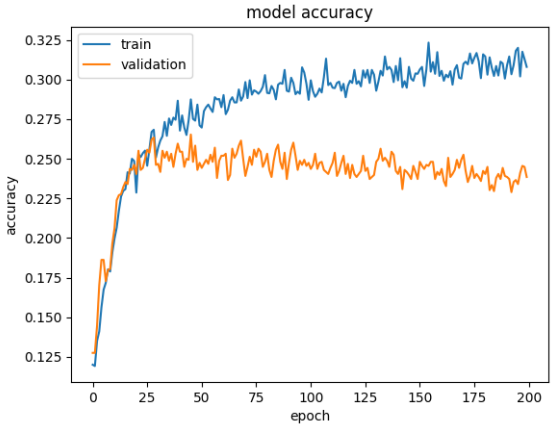

Figure 48: The 2D convolutional neural network for all emotions with normalization and energy, with about a 24% maximum validation accuracy.

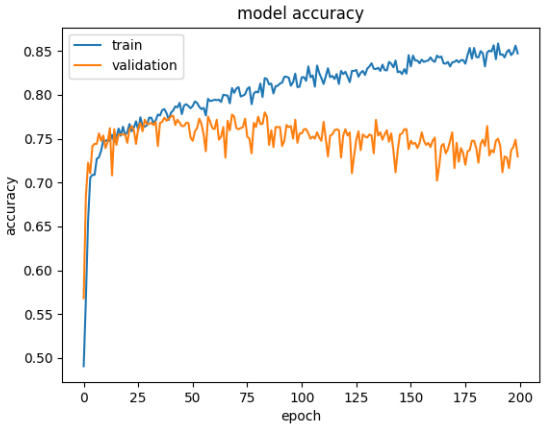

Figure 49: The 1D convolutional neural network for happy and sad with normalization and energy, with about a 75% maximum validation accuracy.

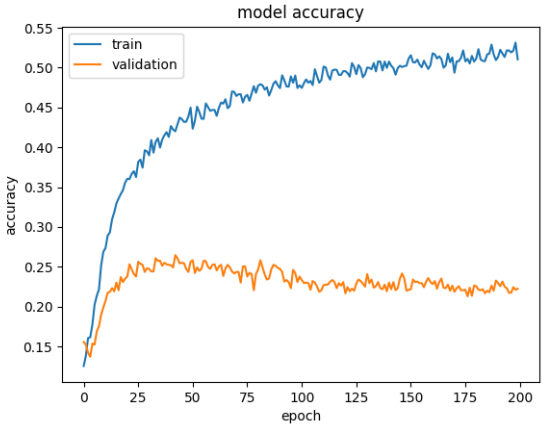

Figure 50: The 1D convolutional neural network for all emotions with energy, with about a 23% maximum validation accuracy.

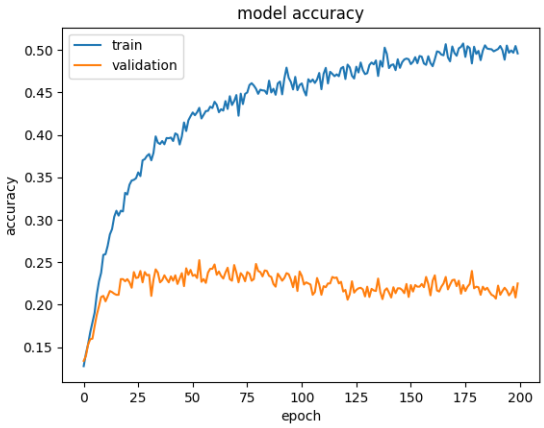

Figure 51: The 1D convolutional neural network for all emotions without energy, with about a 22% maximum validation accuracy.

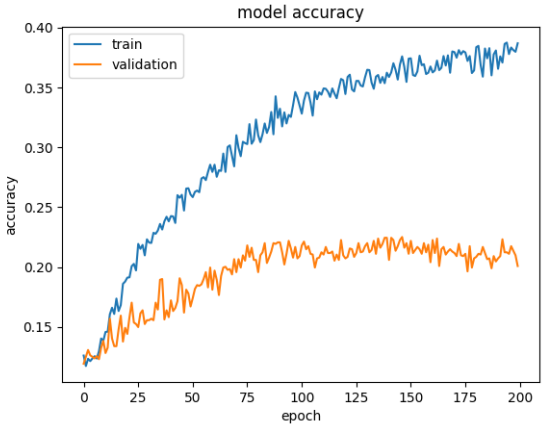

Figure 52: The 2D convolutional neural network for all emotions with energy, with about a 21% maximum validation accuracy.

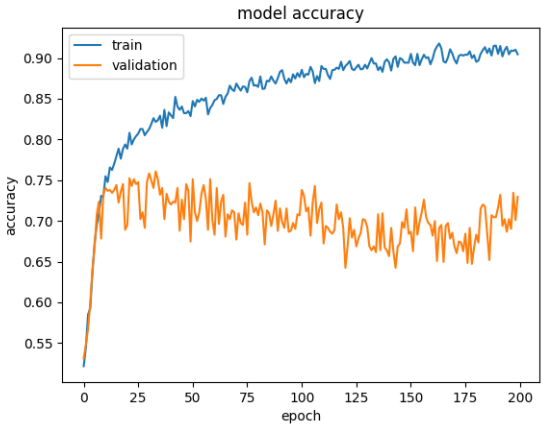

Figure 53: The 1D convolutional neural network for happy and sad with energy, with about a 75% maximum validation accuracy.

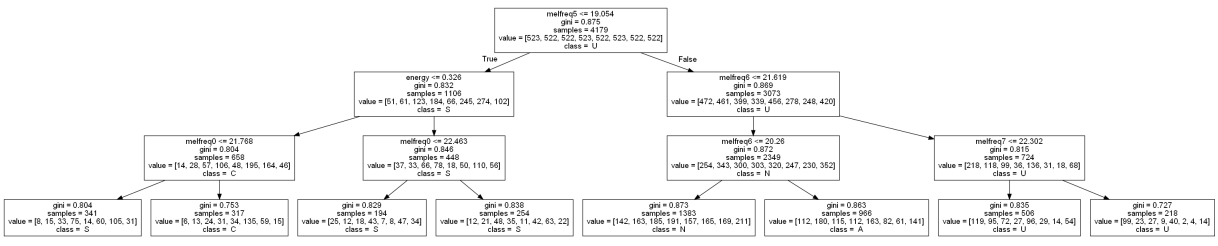

Figure 54: The decision tree for all emotions with depth 3 and an accuracy of 0.200

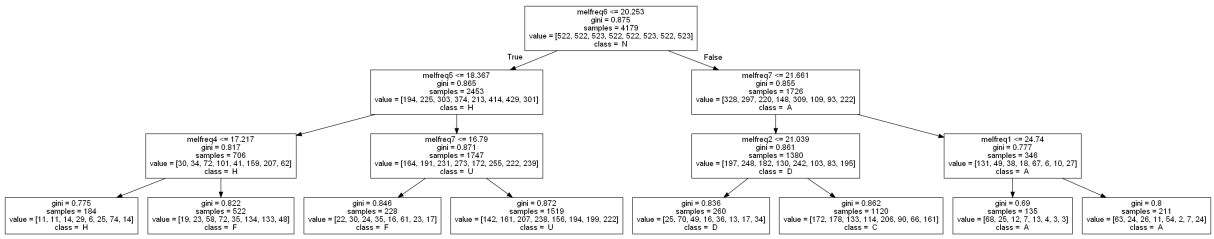

Figure 55: The decision tree for all emotions with depth 3 without energy and an accuracy of 0.208

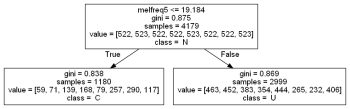

Figure 56: The decision tree for all emotions with depth 1 without energy and an accuracy of 0.189

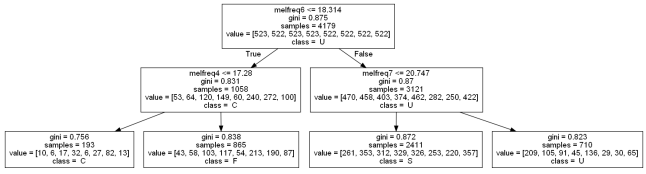

Figure 57: The decision tree for all emotions with depth 2 without energy and an accuracy of 0.1965

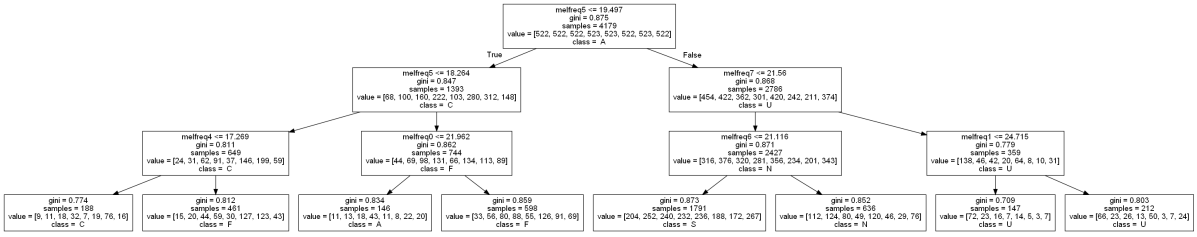

Figure 58: The decision tree for all emotions with depth 3 without energy and an accuracy of 0.204

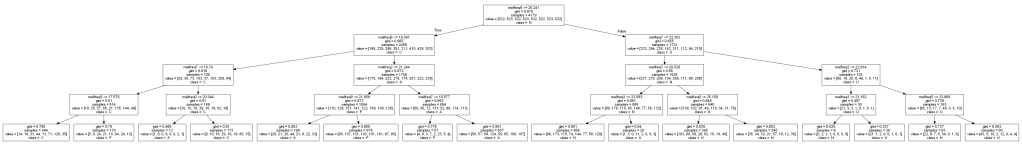

Figure 59: The decision tree for all emotions with depth 4 without energy and an accuracy of 0.230

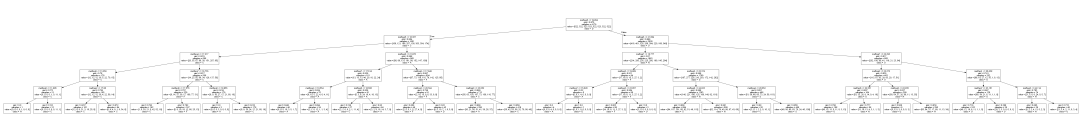

Figure 60: The decision tree for all emotions with depth 5 without energy and an accuracy of 0.233

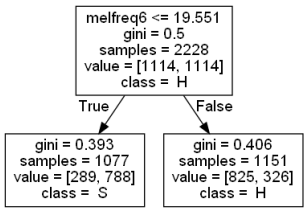

Figure 61: The decision tree for happy and sad with max-depth 5 and an accuracy of 0.728

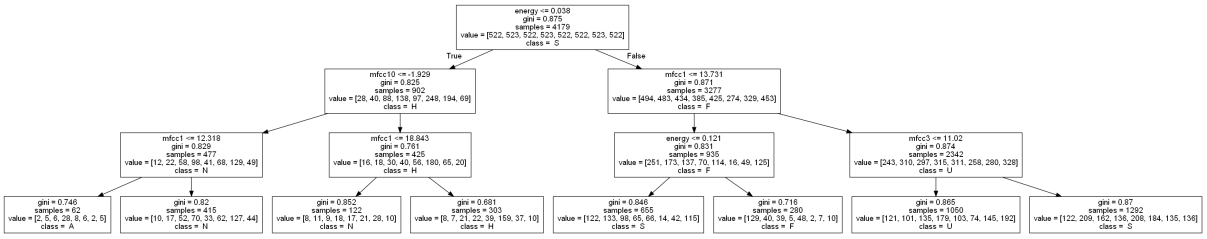

Figure 62: The decision tree for all emotions with depth 3 and an accuracy of 0.203

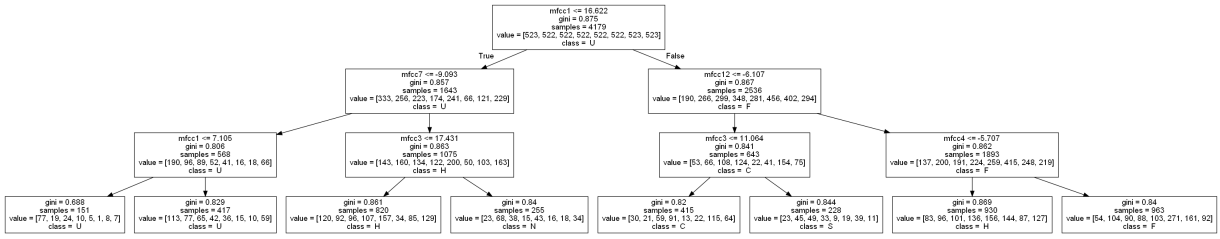

Figure 63: The decision tree for all emotions without energy with depth 3 and an accuracy of 0.215

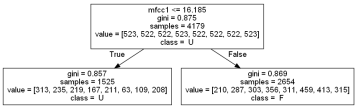

Figure 64: The decision tree for all emotions without energy with depth 1 and an accuracy of 0.189

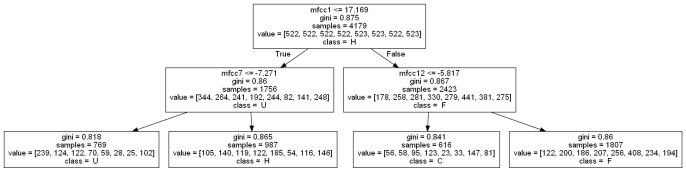

Figure 65: The decision tree for all emotions without energy with depth 2 and an accuracy of 0.210

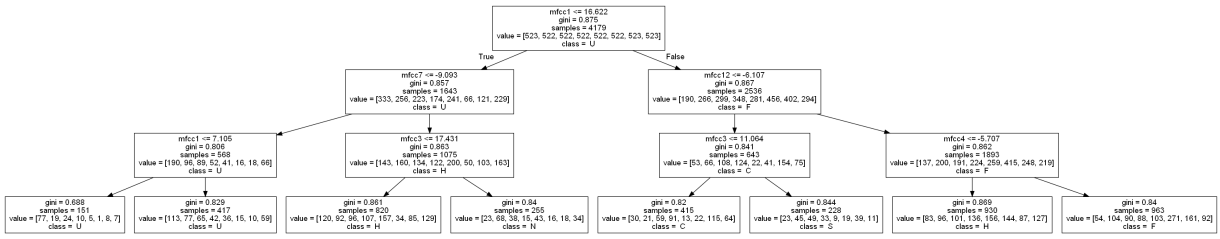

Figure 66: The decision tree for all emotions without energy with depth 3 and an accuracy of 0.226

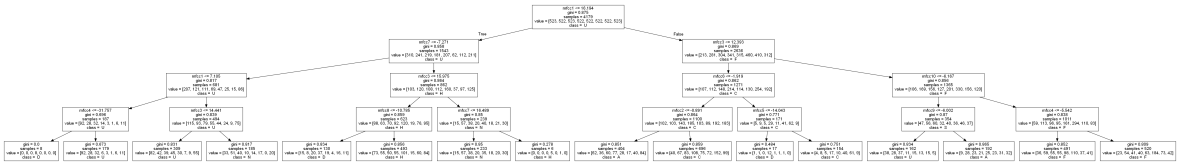

Figure 67: The decision tree for all emotions without energy with depth 4 and an accuracy of 0.249

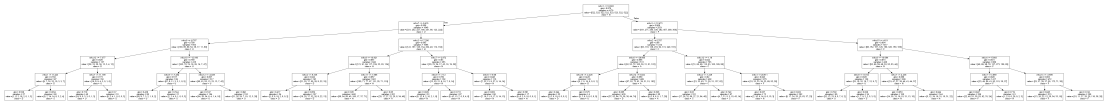

Figure 68: The decision tree for all emotions without energy with depth 5 and an accuracy of 0.253

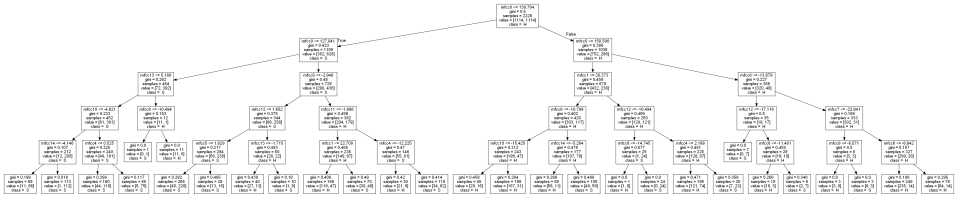

Figure 69: The decision tree for happy and sad without energy with depth 5 and an accuracy of 0.747