Emotion detection from Speech using Neural Networks

Group 16

Mikel Zhobro	Dominik Hollidt	Victor Wolff	Pernilla Wikström
hollidt@kth.se	zhobro@kth.se	vwolff@kth.se	pwikst@kth.se

Abstract

Emotion recognition has a high relevance for dialog managers, which can adjust their response in accordance to the emotion of the speaker. The study investigates the importance of feature extraction in combination with different complex neural network architecture for emotion detection from speech. In order to do so a MLP and a CNN2D were designed by us and compared to an existing CNN2D + LSTM model. Those were trained on a combination of handcrafted features such as MFCC, MEL and CHROMA. We showed that it is possible to achieve accuracies of up to 76% using a CNN2D model in combination with MFCC features on the RAVDESS data set which contains 8 different emotions. The results show that handcrafted feature selection has do be done with much care. Finally the investigation showed that it is even possible to recognize emotions with an accuracy of 63% using raw audio samples only.

Keywords: Emotion recognition, MEL, MFCC, CHROMA, Neural Networks.

1 Introduction

As humans we find speech to be the most natural way to express ourselves. We depend so much on it when we either want a long lasting relationship or a more productive communication in general. It is no surprise that other remedies such as emojis are used to introduce similar effects to other mediums of communications such as emails or text messages. Passing along the emotion is very important to avoid misunderstandings. Emotion also plays a big role when putting emphasis in certain sentences or when trying to convey irony or sarcasm. When responding to a question if the weather is nice, an angry: "yes, it really is", can indicate sarcasm, while a happy response indicates a true statement. Nowadays voice assistance should be able to respond not only to the pure message of a statement, but also to the underlying intent. This requires that emotion detection and its correct handling has to be taught to computers.

First of all it is necessary to define what counts as emotion and what kind emotion exist, since a discretized way of handling of emotion allows for a easier processing and scientific theory. Emotion as a concept is difficult to define and there also exist several different definitions of it in science. Psychology often defines emotion as a complex state of feelings that are based on physical and psychological changes, that affects the behavior and mind. Emotions are basically constituents of syndromes such as motivation, behavior, feelings and physiological changes[5].

One of the most popular literature in speech recognition is Ekman [2] who suggests a list of six discrete emotions consisting of anger, disgust, fear, happiness, sadness, and surprise. According to it, all other emotions can be formed as combination of these base 6 emotions.

With the widespread of speech systems there exist a big potential on emotion recognition technologies which could close the gap between men and machine. Being able to receive and transmit emotional expressions, machines could potentially improve our lives utterly. Nevertheless, similar to all technologies the ethical aspects of emotion technologies should not be underestimated. As it is beautifully presented in the movie HER[1], being able to identify the underlying emotions, AI powered speech systems could possibly learn a lot about the human nature which would enable them skills to even manipulate users. It is therefore necessary to have good ethical intentions when developing these type of systems.

[1] Her, 2013: https://en.wikipedia.org/wiki/Her_(film)

1.1 Our Work

This work does not aim at defining new contexts of emotion or the ethical impact of emotion detection systems, but rather compares how different approaches and features can be used to extract emotions from short speech samples. The results will be presented in the Results section.

Given previous successes of applying deep learning to speech recognition identification tasks, deep learning is a strong candidate for speech-based emotion recognition. In this report we apply deep learning methods to a simple data set featuring 8 emotions. We examine the shortcomings of emotion recognition with MLP and purely CNN based classifiers, and use these findings to motivate models that are better posed to capture temporal dependencies. The novelty of this paper is introduced by examining the accuracy of using stacked MFCC, log MEL and CHROMA features, which are not commonly used for speech emotion recognition. We also designed a pure CNN2D network ourself to compare it to already existing CNNN2D + LSTM networks.

2 Related Work

There has been an increased development of a natural human-computer interface, and thus speech is one of humanity's foremost tools for expressing emotions. It is therefore important to recognize and interpret different emotions in order to respond in a natural way [15].

Using neural-networks for emotion detection from speech requires a careful selection of features, since pattern recognition is rarely independent from its problem domain [3]. Raw audio data [16], Mel-frequency cepstral coefficients (MFCC) [12], just the log-mel spectrograms [16], CHROMA features [13] or LPCC features [5] have been used in emotion detection. It is important to choose features, which also contain potential information about emotion, since pure linguistic features are not of much use for emotion detection [10], which is not directly conveyed in the spoken words.

In addition to the features different models for classifying emotions can be used. In earlier days Hidden Markov Models (HMM) dominated the development of emotional-based speech systems [4]. Nowadays neural networks, which make use of convolutional layers and the LSTM-architecture dominate the field of emotion detection [16, 6].

Experimental studies have shown results with Multi Layer Perceptron (MLP) for speech emotion recognition. Zvarevashe [17] managed to achieve above 70% accuracy using only a MLP on Mel-frequency cepstrum (MFCC) features. These were then extended to more complex models. Wöllmer introduced a LSTM-RNN model and compared it to more conventional support-vector-regression method. The study proved that LSTM-RNN models are superior [14]. Another study done about LSTM is Das [1], where the authors obtain accuracies of about 85% . Zhao et al. extended the LSTM architecture with 1D and 2D convolutional layers to boost up the accuracies to 95% [16]. We have also been heavily inspired by the examples presented in [16], in particular the idea of combining CNNs with a long short-term memory networks (LSTMs).

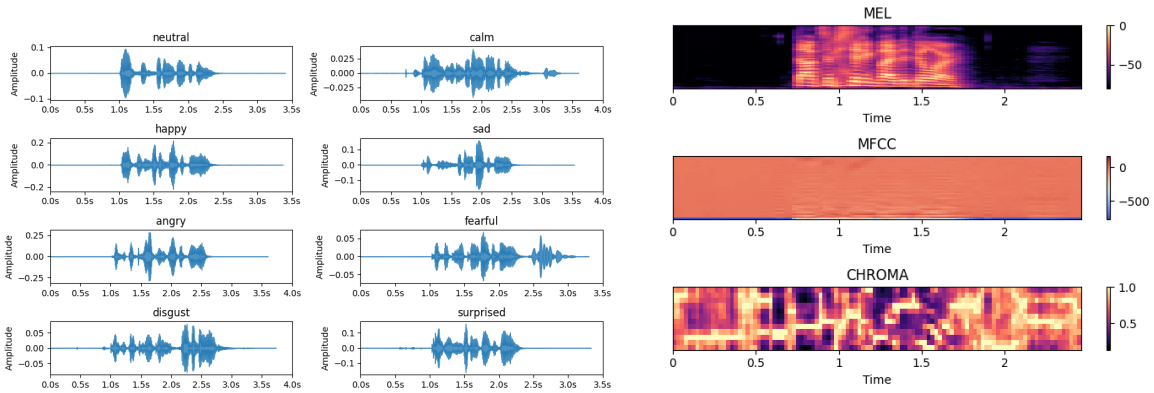
3 Data set

The chosen data set for the task is the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) which is a dynamic, multi-modal set of facial and vocal expression in North American English [9]. In our project we only consider the speech data containing 1440 files with 60 trials per actor $\times$ 24 actors, where female and male actors are equally distributed. The speech samples contains 8 different emotions including calm, happy, sad, angry, fearful, surprise, disgust and neutral. The emotion intensity of each expression is produced at two levels, normal and strong for all emotions beside neutral. This data set is well known for its quality of speakers but does only includes acted emotions which may influence the generalization properties of the trained model. One other problem is the small size of the data set which might not be enough to train complex models with many parameters.

3.1 Preprocessing

As a goal of this investigation we investigate the impact of different features on the accuracy of neural networks. We settled to use (Mel Frequency Cepstral Coefficents) MFCC, MEL and CHROMA features. To allow the extraction and later easy use of features the prepocessing is described in the following.

The idea was to build a dictionary of each speech sample with corresponding emotion and different, interesting features. The raw audio data is converted in frames. Then power spectral density (periodogram) is extracted from the discrete Fourier transforms for each frame. Summing over the discrete bins in them allows to get an idea of the energy content



(a) Audio clips for each emotion

(b) Handcrafted features of a random training data sample.

Figure 1: Pre-proccessed data.

in different frequency regions. Using the MEL-scale we can apply correctly spaced and adjusted filter banks. Once the filterbank energies are obtained, a logarithmic calculation is performed to mimic the non linear loudness understanding of humans. From the just obtained MEL-features it is possible to generate the less correlated MFCC features, by applying a discrete cosine transform [16]. The job of MFCCs is to accurately represent the frame of the short time power spectrum.

In music, the term CHROMA feature or chromagram closely relates to the twelve different pitch classes which are a powerful tool for analyzing music whose pitches can be meaningfully categorized. One main property of CHROMA features is that they capture harmonic and melodic characteristics of speech, while being robust to changes [7] and are thereby suitable for this recognition task.

By using the data set and the python package LibROSA[2] we could easily generate MEL, MFCC and CHROMA features from audio samples with a sampling rate of 16kHz. The result of the preprocessing can be seen in Figure 1b. The subplots display the frames of the speech sample on the x-axis. One can distinguish the frequency content for MEL, MFCC and CHROMA featuers.

4 Method

The basic strategy we follow, to tackle this problem is to gradually increase the model complexity in order to make up for the difficulties of previous models. We start by testing out a simple MLP fed with averaged frequency domain features. The expected loss of information taking place in the averaging process motivates us to try out a 2D Convolution Neural Network (CNN) approach. CNNs are able to learn local features and encode them by using convolution filters. The next logical move lead to the usage of structures that enable us to better capture temporal-dependencies, which is one of the most important aspects in speech. Finally we try out a pure raw audio based method to classify emotions to see how it compares with methods that use handcrafted features as input. The network makes use of 1-dimensional CNN and a LSTM module.

4.1 Multi Layer Perceptron

A simple Multi Lay Perceptron (MLP) has been constructed. It is a feed forward artificial neural network that expects a vector like feature as input. We reduce the second dimension for the 2D features by averaging. In case we want to combine several features they are appended to each other, so that the time dimension stays the same. The MLP consist of an input layer, hidden layers with N number of hidden neurons that uses nonlinear activation functions and the output layer. In this implementation, the ReLu (rectified linear unit) activation was used for the hidden layers. As for the optimization procedure this method minimizes the categorical cross-entropy which enforces a softmax activation on the last layer for treating their outputs as probabilities. The model description can be found in Table 2 in Appendix 8.

[2]LibROSA — librosa 0.7.2 documentation: https://librosa.github.io/librosa/

4.2 Convolutional Neural Network on 2D features

A different approach to classification is to use convolution filters as a mean of storage for local features. In difference to the MLP model, CNN avoids averaging and extracts information directly from the 2D features as illustrated in Figure 2. The first layer tend to detect only simple features which are then combined to encode more specialized features the deeper the neural network gets. The convolution neural network is based on the model presented in [16] and consist of 4 convolution layers with 16,32,64 and 128, 3x3 filters respectively. Only the first convolution layer is followed by max-pooling with filter size 2x2 and stride 2, the rest use max-pooling filters of size 4x4 and stride 4. The neural network is concluded with a fully connected layers of size 512 and the classification layer. All layers are ReLu activated. The neural network is optimized using categorical cross-entropy loss function with softmax activation function in the classification layer. The exact network setup can be found in Table 3 in Appendix 8.

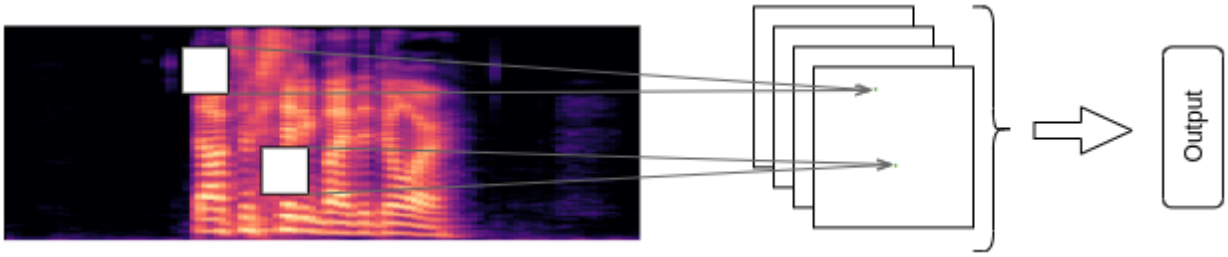


Figure 2: Example of 2D CNN on MFCC features

4.3 Convolutional Neural Network and Long Short Term Memory

In this approach we build on the method presented in [16] to do emotion recognition using a CNN followed by a LSTM. The motivation behind this combination is to take advantage of the strengths of both networks and overcome their respective shortcomings. The CNN plays the role of local feature extractor whereas the LSTM learns the global features by capturing the long-term contextual dependencies. This method can be used both on 1D input features, such as raw audio signals or 2D handcrafted features such as LMFCC, CHROMA or MEL, see Figure 2. While the authors in [16] use only MEL features in this project we experiment with different combination between LMFCC, CHROMA and MEL.

In more details, the CNN is build from several so called local feature learning block (LFLB) which consist of a convolution layer with batch normalization and ELU activation, followed by a max-pooling layer.

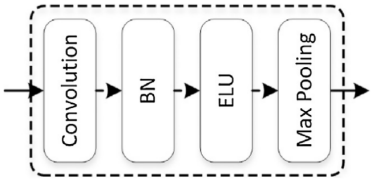


Figure 3: local feature learning block (LFLB).

Batch normalization is used to normalize the activations from the convolution layer at each batch by applying a transformation that maintains the mean activation close to 0 and the activation standard deviation close to 1. The normalized features coming from the CNN are then activated through ELU activation function

$$\sigma(x) = \begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

with α being a constant positive number ($\alpha > 0$). The max-pooling layer performs down sampling in order to reduce the resolution of the features and generalize the results from convolution filter making the detected features invariant to scale and orientation. The features produced by max-pooling can be expressed as:

$$z_k^l = \max_{\forall p \in \Omega_k} z_p^l$$

where Ω_k is the pooling region with index k.

For both 1D and 2D case the network has the same structure. It is compounded of 4 LFLBs, which output is then reshaped and fed to the LSTM and then classified with a fully connected layer activated with softmax which enables a probabilistic representation of the output.

All used convolution filters have a 3x1 and 3x3 size for the audio clip and 2D features respectively. The number of filters for each LFLB are 32,64,128 and 256 for the raw audio signals and 64,64,128 and 128 for the 2D features. For the first case we use only max pool filters of size 4 and stride 4. For the second case, beside the first blocks which uses pooling and stride of size 2 all other blocks consist of pooling filters of size 4 and stride 4. The used LSTM layers have a hidden state of dimension 512 and 256 for the raw audio signal and 2D features respectively. The overall architecture of the CNN LSTM networks is illustrated in Figure 4

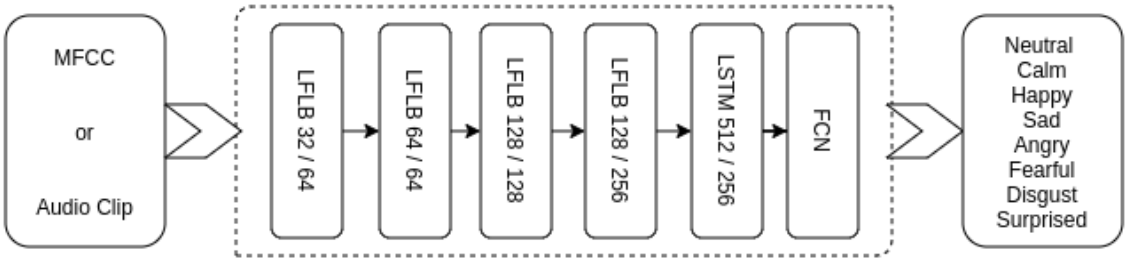


Figure 4: local feature learning block (LFLB).

Another important design decision is the reshaping that takes place just before the LSTM. In order to preserve the temporal dependencies of the local features we reshape the output by keeping as feature dimensionality the number of filters of the last LFLB block. For the case of raw audio signals a graphical interpretation is given in Figure 5. Features are represented with a specific color.

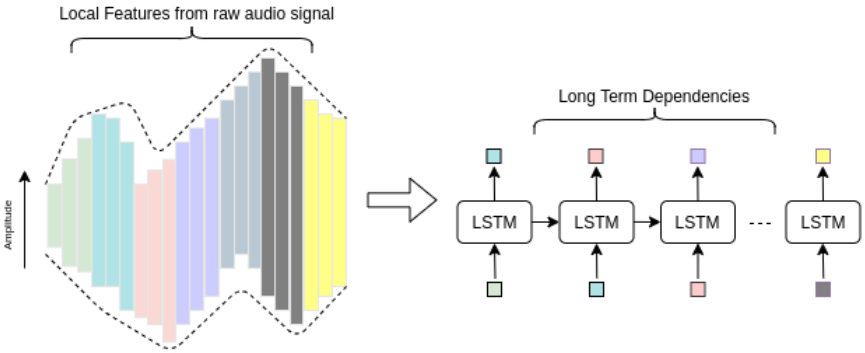


Figure 5: local feature learning block (LFLB).

Similar to above, after learning local features from MFCC features (illustrated in the left side of Figure 6) they are reshaped to form a temporal sequence as illustrated in Figure 6. The sequential order is left-right and bottom-up. The initial hidden state is chosen randomly and each learned local feature is represented with a certain color. We see that once again the dimensionality of the features correspond to the number of filters of the last LFLB block.

5 Experiment

The evaluation on multi-class classification problems is simple in general. The data is partitioned into training, validation and test sets. The models are trained on the training set and the validation set is used to tune the hyperparameters. The performance of the models is measured with means of confusion matrix. The approach consist of finding the number of correct and incorrect predictions for each of the classes. Thus, the results give an insight on the errors as well as the classes where they occur. Furthermore, we investigate how well the networks are able to fit the eight labels by comparing them with a reduced label set of four. These 4 emotions (anger, sadness, happiness and fear) have been most used in other data sets and make comparison easier [11].

In addition to that, we decide to consider robustness of the models in the evaluation by performing K-fold crossvalidation as it presents the results by a less biased and less optimistic estimation of the model skill and at the same time considers

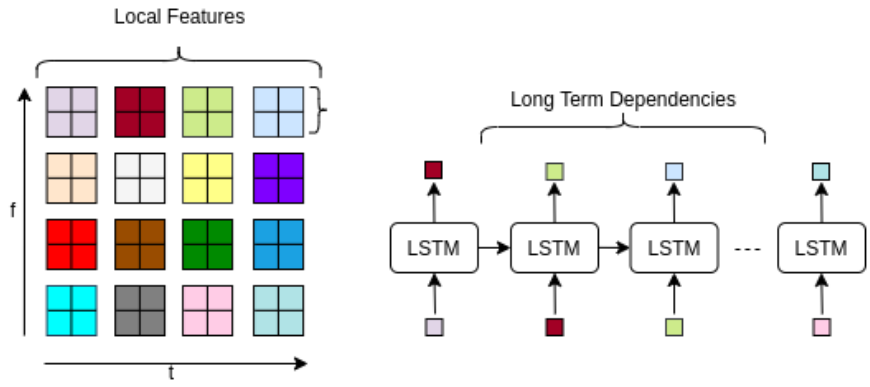


Figure 6: Sample figure caption.

the generalization properties of the model. The idea is to split the training set into K groups which are sequentially hold out for validation while the remaining is used for training. At the end, the accuracy is computed by averaging the accuracies from each fold. We exploit an *Early stopping* technique to stop the training iteration of fold k if there has been no improvement of the validation set for a specified number of epochs. Hence, number of epochs will be handled automatically.

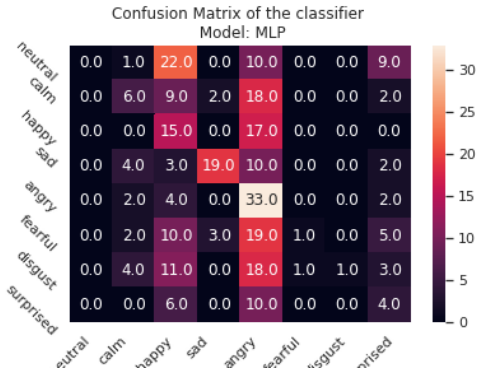
It is important to mention that the models used in K-fold crossvalidation come with tuned hyperparameters. For example all models are tuned for the type optimizer(rmsprop, adadelta, adam). MLP and CNN2D perform best with adam whereas CNN2D LSTM with rmsprop. In addition to that we tune an extra dropout layer for the CNN1D LSTM before the classification layer. We found out that a dropout of 0.25 greatly robustifies the model against overfitting compared to the original model. Finally, the CNN2D model is hypertuned on the number of layers and number of filters for each layer. We used Hyperband [8] to select hyperparameters for each model.

For the implementation of our models we use the Keras framework with Tensorflow backend. The confusion matrices are calculated with the help of sklearn library. The data is normalised to have zero mean and a standard deviation of one. In the following we present the K-fold cross validation plot only for the MLP case. For the rest of the models we only show plots of the last fold, for clarity reasons. The resulting accuracy of each network are shown in Table 1. The table includes the performance on different selections of features and emotions.

Table 1: Final accuracy of emotion recognition systems

Dataset	4 Emotions		All Emotions	
Model	Val. acc.	Test Acc.	Val. acc.	Test Acc.
MLP with MFCC, MEL & CHROMA	55.88%	55.19%	35.50%	32.92%
MLP with MFCC & MEL	50.82%	47.73%	36.63%	33.92%
MLP with MFCC	51.50%	51.49%	34.47%	36.22%
CNN2D with MFCC, MEL & CHROMA	88.31%	85.06%	75.01%	71.34%
CNN2D with MFCC & MEL	87.66%	86.36%	73.64%	73.86%
CNN2D with MFCC	92.86%	89.11%	78.22%	76.44%
CNN2D + LSTM with MFCC, MEL & CHROMA	81.17%	80.74%	69.91%	66.8%
CNN2D + LSTM with MFCC & MEL	83.77%	79.84%	72.92%	68.77%
CNN2D + LSTM with MFCC	88.94%	87.12%	80.02%	79.442%
CNN1D + LSTM with raw audio samples	77.27%	76.55%	64.12%	63.54%

The confusion matrix of the MLP fed with 8 emotions using the MEL, MFCC and CHROMA features is seen in Figure 7a. Furthermore, the cross validation of the loss and accuracy over epochs is seen in Figure 7b.



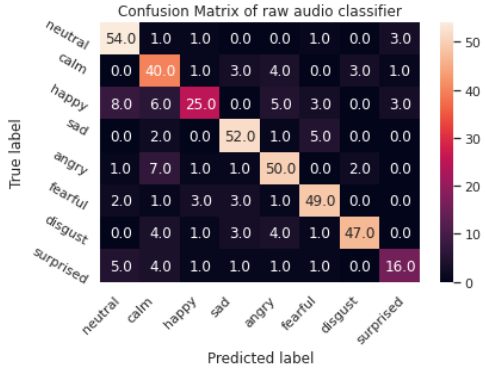
(a) Confusion matrix

(b) Loss and accuracy with a dashed and solid trajectory for validation and training set respectively.

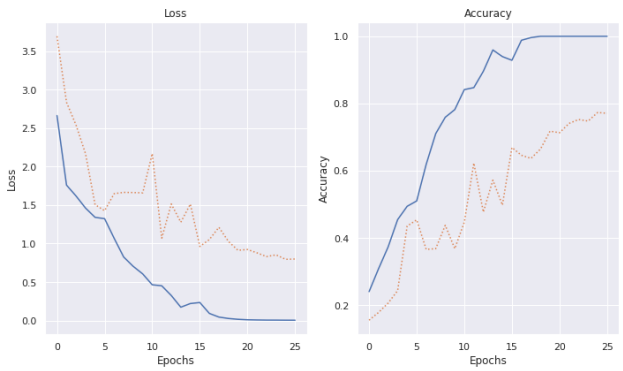
Figure 7: Results of the MLP on MEL, MFCC and Chroma features

The performance of the CNN model on MFCC features is given by the confusion matrix in Figure 8a.

The accuracy and loss of the last fold in cross validation is illustrated in Figure 8b, with a dashed and solid trajectory for validation and training set respectively.



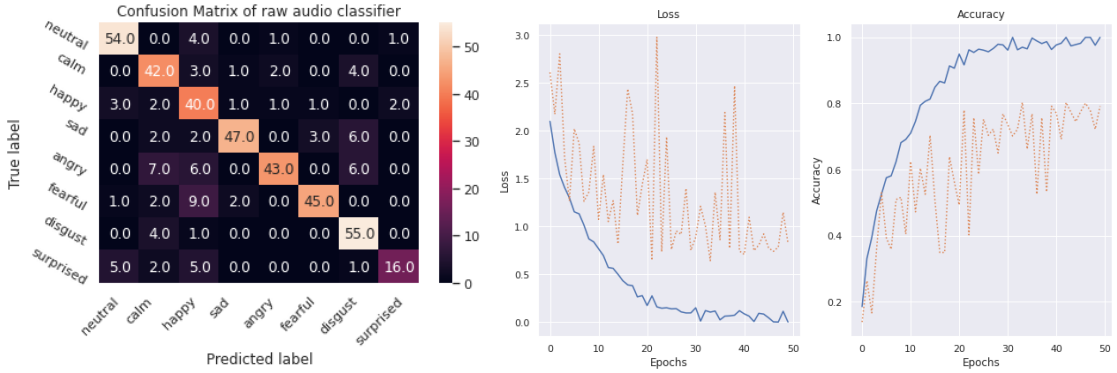
(a) Confusion matrix



(b) Loss and Accuracy over epochs

Figure 8: Results of the CNN2D on MFCC features

Figure 9 shows the results of the CNN-LSTM network using MFCCs. Cross validation accuracy and loss over epochs is seen in Figure 9b, for the validation and training data respectively.

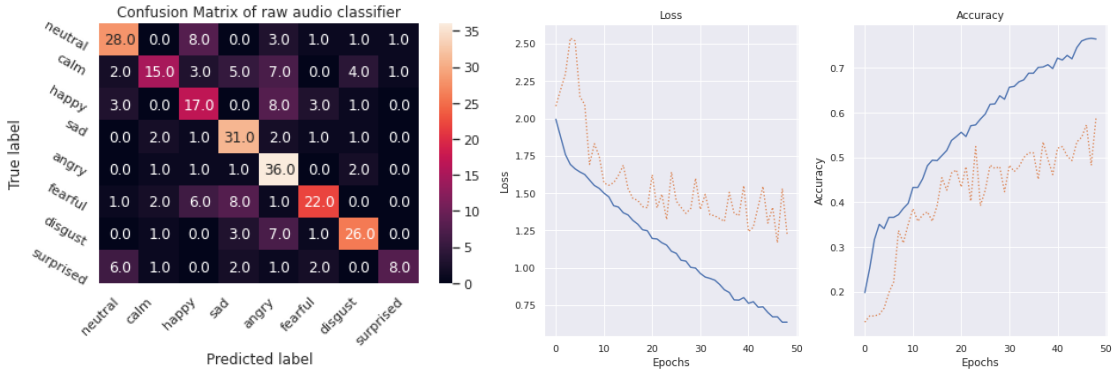


(a) Confusion matrix

(b) Loss and Accuracy over epochs

Figure 9: Results of the CNN2D + LSTM on MFCC features

Finally, Figure 10 shows the performance of the CNN-LSTM trained on raw audio and its loss-accuracy plots.



(a) Confusion matrix

(b) Loss and Accuracy over epochs

Figure 10: Results of the CNN1D + LSTM on raw audio data

6 Discussion

Our initial experiments using a very simple MLP, proved that it is not possible to accurately detect emotions from speech using averaged features. In Table 1 it is clear to see that the simple MLP only achieves accuracies of 37% or less for all emotions. Reducing the emotions set to four shows an increase of accuracy to around 50%, which is still not good considering that randomly guessing would yield a accuracy of 25%. Looking at the confusion matrix in Figure 7a shows that the network is not able to learn emotions properly and mainly guesses a happy or angry emotion. The other emotions are almost never predicted. For the MLP network a very high variance in accuracy throughout the learning process can be observed. This indicates not very robust features. This can be caused by the averaging of the features. Finally it is noticeable that the accuracy decreases with increasing number of features using all emotions as a label. For the reduced label set the trend is not really identifiable. In total a simple MLP model with a single hidden layer is not suited for a robust and accurate speech emotion recognition.

By using convolutional neural networks to extract features, we are able to dramatically increase accuracy for both only 4 emotions and all emotions as shown in Figure 8b. One reason for that may be the preservation of 2D features. In addition to that it is noticeable from the Table 1 that using more than one feature doesn't improve the performance necessarily. For example, using MEL or both MEL and CHROMA features in addition to MFCC results in worse accuracy in general. This result is not expected in a first view, since the MFCC is already included in all cases. There are two reasons that can explain this occurrence. CNNs in general expect uniform input data to learn from. Learning

kernels from input features of different types simultaneously may present a big limitation. Furthermore, the small size of the data itself puts a constraint on the complexity of the patterns that can be learned which may influence the learning of more complicated patterns on the border between two types of features. Another general limitation of this approach lies on the fact that CNNs learn only local features of the input data and are unable to capture long temporal dependencies.

But since speech signals represent a sequential data type it is intuitively reasonable to extend the model with components that capture the temporal dependencies. Nevertheless using CNN LSTM does not exactly lead to any improvements and actually performs slightly worse than the CNN2D model in our experiments. This can be an indication that 2D CNN are already learning the temporal context to a degree since the convolution filters slide on both temporal and feature dimensions. Another reason can be the overall constellation of how we input CNN-learned features in the LSTM which may results in learning temporal dependencies that are not very crucial for emotion classification.

Using raw audio samples instead of handcrafted features shows reasonable yet slightly lower accuracy. The CNN1D + LSTM model produces around 10% lower accuracy than the CNN2D models for both the all emotions and four emotions set. The confusion matrix in Figure 10a shows that the network generally learned the emotions, but has some missclassfications especially for calm and fearful. The learning rate suggest that the accuracy could be improved by having a bigger data set available, since the loss and accuracy did not converge yet. On the other hand we see a divergence in validation and training loss and accuracy, which indicates a overfitting of the training data. Overall the network proved to be able to extract relevant features from the harder to classify raw data. It can be assumed that using a more complex model in addition to more training data can achieve similar accuracies to models using handcrafted features.

The performance of the networks was quite accurate even though the split of the data set could have been more balanced among train, test and validation set in regard to emotion labels, The choice of the reduced data set showed that the emotions have well defined differences to achieve better accuracy, which is understandable due to the fewer number of classes.

7 Conclusion

In this work we successfully build models to recognize emotions from the RAVDESS data set. We experimented using different network models and analysed their performance based on handcrafted features, a combination of handcrafted features and raw audio data. The results showed that an increase on the number of features did not result in higher accuracy. Additionally we found out that in our case, a combination of a CNN and a LSTM network did not bring any improvement over a less complex CNN model. Finally the results verify that both implementations of handcrafted features and raw audio files can be used to recognize all 8 emotions for the given data set.

References

[1] Asit Kumar Das, Janmenjoy Nayak, Bighnaraj Naik, Soumi Dutta, and Danilo Pelusi. *Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2020*. Springer Nature, 2020. Google-Books-ID: nrPRDwAAQBAJ.

[2] Paul Ekman. An argument for basic emotions. *Cognition and Emotion*, 6(3-4):169–200, May 1992. Publisher: Routledge _eprint: https://doi.org/10.1080/02699939208411068.

[3] Moataz El Ayadi, Mohamed S. Kamel, and Fakhri Karray. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572–587, March 2011.

[4] Zeynep Inanoglu and Ron Caneel. Emotive alert: HMM-based emotion detection in voicemail messages. In *Proceedings of the 10th international conference on Intelligent user interfaces - IUI '05*, page 251, San Diego, California, USA, 2005. ACM Press.

[5] Leila Kerkeni, Youssef Serrestou, Mohamed Mbarki, Kosai Raoof, Mohamed Ali Mahjoub, and Catherine Cleder. Automatic Speech Emotion Recognition Using Machine Learning. *Social Media and Machine Learning*, March 2019. Publisher: IntechOpen.

[6] 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. Conference Name: IEEE Access.

[7] Ted Kronvall, Maria Juhlin, Johan Swärd, Stefan I. Adalbjörnsson, and Andreas Jakobsson. Sparse modeling of chroma features. *Signal Processing*, 130:105–117, January 2017.

[8] Lisha Li, Kevin G. Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Efficient hyperparameter optimization and infinitely many armed bandits. *CoRR*, abs/1603.06560, 2016.

[9] Steven R. Livingstone and Frank A. Russo. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS), April 2018. type: dataset.

[10] Björn Schuller, Anton Batliner, Stefan Steidl, and Dino Seppi. Recognising realistic emotions and affect in speech: State of the art and lessons learnt from the first challenge. *Speech Communication*, 53(9):1062–1087, November 2011.

[11] Dimitrios Ververidis and Constantine Kotropoulos. A Review of Emotional Speech Databases. page 15.

[12] Vishal B Waghmare, Ratnadeep R Deshmukh, Pukhraj P Shrishrimal, and Ganesh B Janvale. Emotion Recognition System from Artificial Marathi Speech using MFCC and LDA Techniques. page 10.

[13] Sifan Wu, Fei Li, and Pengyuan Zhang. Weighted Feature Fusion Based Emotional Recognition for Variable-length Speech using DNN. In *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*, pages 674–679, June 2019. ISSN: 2376-6506.

[14] Martin Wöllmer, Florian Eyben, Stephan Reiter, Björn Schuller, Cate Cox, Ellen Douglas-Cowie, and Roddy Cowie. Abandoning Emotion Classes – Towards Continuous Emotion Recognition with Modelling of Long-Range Dependencies. pages 597–600, January 2008.

[15] Yue Xie, Ruiyu Liang, Zhenlin Liang, and Li Zhao. Attention-Based Dense LSTM for Speech Emotion Recognition. *IEICE Transactions on Information and Systems*, E102.D(7):1426–1429, July 2019.

[16] 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, January 2019.

[17] Kudakwashe Zvarevashe and Oludayo Olugbara. Ensemble Learning of Hybrid Acoustic Features for Speech Emotion Recognition. *Algorithms*, 13(3):70, March 2020. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

8 Appendix

Table 2: Model structure of MLP.

LAYER TYPE	OUTPUT SHAPE	ACTIVATION
Dense	(None,300)	ReLu
Dense	(None, $N_{\text{target classes}}$)	Softmax

Table 3: Model structure of CNN on 2D features.

LAYER TYPE	SIZE AND/OR FILTERS NUMBER/STRIDE	ACTIVATION
Conv2D	3x3, 16	ReLU
MaxPooling2D	2x2, 2	-
Conv2D	3x3, 32	ReLU
MaxPooling2D	4x4, 4	-
Conv2D	3x3, 64	ReLU
MaxPooling2D	4x4, 4	-
Conv2D	3x3, 128	ReLU
MaxPooling2D	4x4, 4	-
Flatten	-	-
Dense	512	ReLU
Dense	8	Softmax