# Effect of Feature Extraction when Classifying Emotions in Speech

An Applied Study

*Henrik Holmström*
*Victor Zars*

**Henrik Holmström**
**Victor Zars**
VT 2018
Examensarbete, 15 hp
Supervisor: Eddie Wadbro
Examiner: Juan Carlos Nieves Sanchez
Kandidatprogrammet i datavetenskap, 180 hp

## Abstract

The demand for machines that can interact with its users through speech is growing. For example, four of the world's largest IT companies; Amazon, Apple, Google and Microsoft, are developing intelligent personal assistants who are able to communicate through speech. In this thesis, we have investigated the effect of feature extraction when classifying emotions in speech, using a convolutional neural network (CNN). We used the software openSMILE to extract two sets of features, and one set of raw data, from recorded audio, and compared the CNN's classification accuracy of the sets with eight, five and three classes of emotions. We used one architecture of the CNN, to be fair when comparing each feature set, and implemented it using Keras. The CNN architecture was developed by an experimental approach.

The feature set that gave the highest accuracy managed to reach 39 % accuracy when classifying eight emotions (random guessing would yield around 12.5 % accuracy on average), 53 % with five emotions (compared to around 20 % if just guessing), and 69 % with three emotions (compared to around 33 % if just guessing). This set also performed best when distinguishing emotions from each other.

The result shows that using feature extraction improves the accuracy, but more features does not necessarily increase accuracy.

While the classification accuracies in this study may seem low, it is important to remember that even for humans, it can be hard to distinguish different feelings based on just the pitch of other people's voices.

## Acknowledgments

We would like to thank our supervisor at Umeå University, Eddie Wadbro, for valuable guidance through the project.

We also want to thank Jon Granberg and Mattias Gunnarsson at Acino, and Joakim Sundqvist and Peter Vestman at Försäkringsfabriken, for bringing up the idea for this thesis.

We would like to express our gratitude to Olle Sundin for answering our question and bringing valuable input throughout the project.

## Contents

# 1 Introduction

The background that motivates this study, as well as some earlier work within the field of *emotions in speech classification* will be presented in this section. A research question will be presented, and a discussion hold about potential ethical aspects that needs to be taken into consideration.

## 1.1 Background

Common computer and smartphone users encounter artificial intelligence (AI) on a regular basis, for example, in web based customer support, and it will probably not be decades before robots play a central part of the residential care.

In the article *Autonomous Systems to Support Social Activity of Elderly People - A Prospective Approach to a System Design*, Reis et al. [1] present suggestions on how autonomous intelligent systems can keep old and lonely people stimulated and help them to stay in touch with their family and/or friends. The intelligent assistant should be able to recognize the emotional state of its user, and based on that information propose suitable activities, such as greetings, social media management, social event management and social games. Reis et al. [1] proposed to use image analysis techniques for user identification and state of mind assessment. However, it is not the only possible technique available.

In a later article by Reis et al. [2], *Using Intelligent Personal Assistants to Strengthen the Elderlies' Social Bonds - A Preliminary Evaluation of Amazon Alexa, Google Assistant, Microsoft Cortana, and Apple Siri*, the authors state that *"Social isolation and loneliness are among the important factors for the degradation of the life quality as the persons' aging process advances."* and they evaluate how well some of the voice controlled electronic intelligent assistants already on the market can be used to keep and strengthening the bonds between elderly people and their families and friends. Although none of the electronic intelligent assistants offered all the requested features mentioned earlier, the study shows that electronic intelligent assistants have great potential.

Though many computer applications and robots today are "smart", users sometimes get frustrated when they feel that the robots can not see things in the users' context and act accordingly. The users will perceive the interaction as more smooth and most likely in a more positive way if the robot adjusts its interaction based on the users' emotions. The voice mode is one important factor in determining the users' emotions. Other important factors are the users' facial expressions and body posture.

For user-interactive artificial intelligence to become relevant in everyday situations, it has to be able to understand what the user really means, not only what the user says. In order for such applications to adapt their responses to their user's feelings, the applications must be able to determine the user's emotional state.

This study will focus on how to determine the emotional state of a user based on his or her voice mode, i.e., not by what the user says, but by *how* it is said. So, how do you write a program that can classify emotions based on audio streams? In this thesis, we investigate if machine learning (ML) can answer this question. ML is a field within computer science dealing with computer systems that perform statistical analyzes on task specific data to find general patterns to be able to answer a specific task.

### 1.2 Related Work

By using a deep convolutional neural network for facial recognition, a deep belief net to capture audio information, a deep autoencoder to model the human actions and a shallow network architecture to extract features from the mouth of the primary humans in the scene, Kahou et al. [3] managed to reach an accuracy of 41.03 %.

Mena [4] compared sequential minimal optimization (SMO), naive bayes, and logistic model tree (LMT) classification using three different feature sets created by the software openSMILE. They managed to get 86 % accuracy for SMO, 75 % for Naive Bayes and 83 % for LMT. They then further the comparison by applying a feature selection techniques and comparing them. They concluded that having a higher number of features did not always justify the usage, but rather having important features brings the same or better accuracy.

Zheng, Yu, and Zou [5] experimented using a deep convolutional neural network (DCNN) with preprocessed speech signals. They use principal component analysis (PCA) on the signals to reduce the dimension. Then they further split the signals into non-overlapping segments. They then train a DCNN that consists of two convolutional and two pooling layers and achieve about 40 % classification accuracy when classifying five emotions. They also said that the resulting DCNN outperforms support vector machine (SVM) based classification on same hand-crafted features.

Sundin [6] used a convolutional neural network (CNN) to classify bird song to study the effects of noise reduction. The raw data compared to preprocessed data had roughly a difference of 17 %. They concluded that any model trained on data with any degree of noise reduction had higher classification accuracy than models trained on data without noise reduction.

Lin and Wei [7] used hidden Markov model (HMM) and support vector machine (SVM) with extracted features to recognize emotions in speech. By first extracting a few selected features to be used as input for the HMM classifier and compare the accuracy with the accuracy of HMM with Mel frequency cepstrum coefficients (MFCC) they concluded that ”while MFCC are popular features in speech recognition they are not suitable for emotion recognition in speech”. For the SVM classifier they used a different set of features which can be an indicator of emotional states in speech. They compared the accuracy of SVM with the accuracy of k-nearest neighbors (KNN) which showed that SVM outperformed KNN with an accuracy of 88.9 %.

Abdel-Hamid et al. [8] showed that using a 1-D convolutional neural network with limited weight sharing reduces the error rate by 6-10 % for speech recognition. The feature vectors are generated using Fourier-transform-based filter-bank analysis, which includes 40 log energy coefficients distributed on a mel scale, along with their first and second temporal derivatives.

### 1.3 Research Question

This is an applied study with the purpose to look into;

> *”How does different feature extractions sets affect accuracy when classifying emotions in speech using a convolutional neural network?”*

The reason a CNN was chosen as model is that a CNN is good when the structure of the data is important, which it is in the case of raw data, where the data can be seen as a spectrogram. However, when using feature extraction, the structure of the data is not important, as long as the data for each audio sample has the same structure. Since the main focus is to compare and evaluate if/how well feature extraction configurations can improve the perfromance, we will use a CNN for both raw data and data extracted by a configuration tool. If two different models would have been used for raw data and feature extraction configurations, the results might be more affected by the model than the usage of feature extraction.

### 1.4 Ethical Aspects

There are some ethical aspects to consider. For example, the CNN in this study will be trained and tested on audio files containing sentences in English. Since different languages and dialects can sound very different acoustically, it is possible that the network only will be able to produce accurate predictions on the language and dialects it has been trained on.

When using emotion recognition inside an application, incorrect feedback can be given to the user, due to the fact that the classifying model can be wrong. This can result in a negative user experience that can lead to frustration, and in the worst case scenario, users being offended.

## 2 Theory

In this section, a brief introduction to the theories used in this study are presented, including machine learning, artificial neural networks, convolutional neural networks, feature extraction and, confusion matrices.

### 2.1 Machine Learning

Machine learning (ML) is a field within computer science dealing with computer models that uses statistical methods to learn how to recognize patterns from data, and then give it a classification or give a recommendation based on that information, without being pre-programmed for a specific task.

### 2.2 Artificial Neural Network

An artificial neural network (ANN) is a ML model that, by analyzing examples of data tries to find a function that maps the input data to its correct output data, instead of using pre-defined rules. This type of model is often trained under the principle of supervised learning, with known input-output-pairs. An ANN can for example be used to classify images, or recommend the next move in a game.

An example of how a simple ANN can be structured and work, for a multi-class classification problem of two classes, that is trained under the principle of supervised learning, will now be presented. In supervised learning, the true label of the input data is known, and the model is trained to be able derive that output for unlabeled data.

An ANN consists of nodes, also known as artificial neurons, which are inspired by the biological neural network in a brain. The nodes are ordered in layers of three main categories; first one *input layer*, followed by one or multiple *hidden layers*, and finally one *output layer* (see Figure 1).

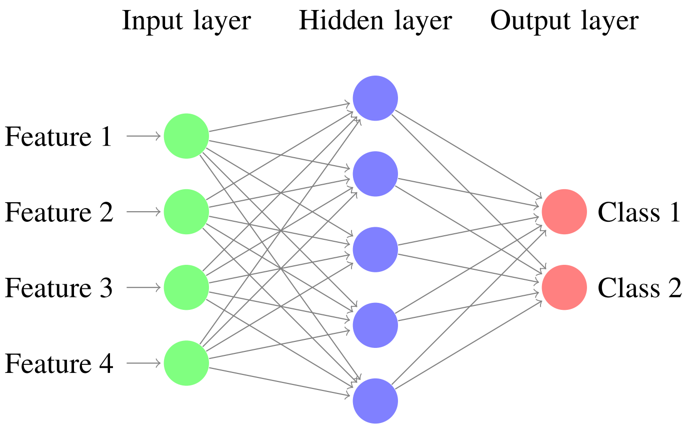


**Figure 1:** An artificial neural network (ANN) with one hidden layer. The two nodes in the output layer represent one class each in a multi-class classification problem.

The information above about an ANN in Section 2.2, was gathered from a tutorial video by deepLizard, called *Artificial Neural Networks explained*, from a video serie called *Machine Learning & Deep Learning Fundamentals*, uploaded on YouTube by the user deepLizard [9].

The following information presented in Section 2.2 is also gathered from the same tutorial video serie.

### Input layer

Every node in the input layer represents an individual feature from an input data sample to the network, and are connected to either one node, some nodes, or every node in the first hidden layer. Each connection has an individual weight, with an assigned value between zero and one, which represent how strong each connection between the nodes are. The weights are usually initialized with small random value. An individual feature is represented by a numeric value.

### Hidden layer

The input to each node in the hidden layer, is calculated by the propagation function; a weighted sum of each of the features of each of its connected nodes in the previous layer, passed through a strictly monotone increasing *activation function*, that produces av value representing the stimuli to that node, just like neurons get activated in the brain (see Figure 2).

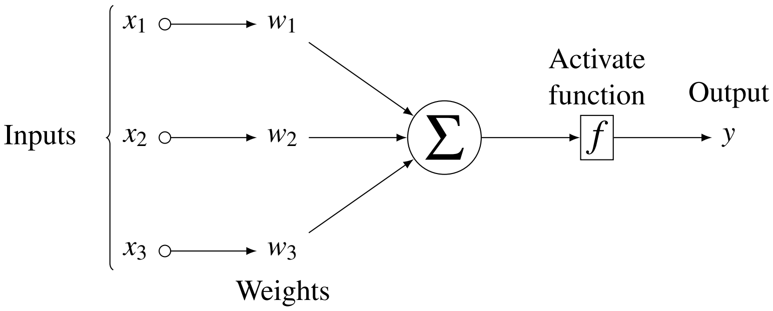
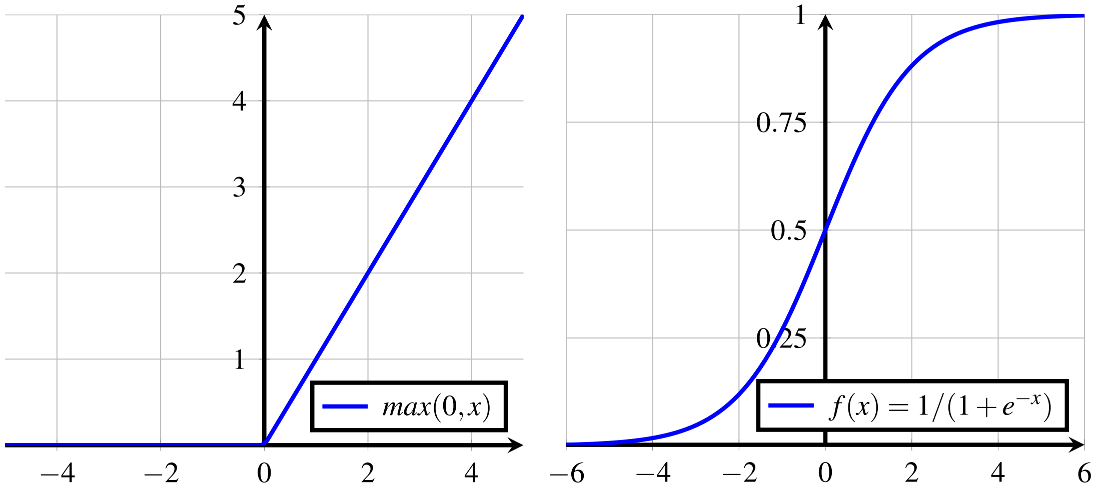


**Figure 2:** A visualization of the propagation function.

Two common activation functions are the rectified linear unit (ReLU) and sigmoid (see Figure 3). This process of activation stimuli is run through the whole network, from the input layer to the output layer, and is called *propagation*.



**(a)** ReLU

**(b)** Sigmoid

**Figure 3:** Two common activation functions; rectified linear unit (ReLU) and sigmoid. The sigmoid (Figure 3b) function maps its input value between zero and one, where one means full activation and zero means inactivated. The ReLU (Figure 3a) function maps its input value to zero if it is negative, or the input value itself.

### Output layer

The output layer is a vector that, in a multi-class classification problem, for each possible class, represents the likelihood that the input data belongs that class.

### Training an ANN

The training process of an ANN is to optimize the weights of the connections between the nodes, to map the input data to its correct output label. To calculate how the weights should be adjusted, an *optimizer* whose goal is to minimize a *loss function*, which is a metric of the error between the predicted class (in a classification problem) and the true class (known by its label in supervised learning) is used. A common optimizer is the stochastic gradient descent (SGD), and two common loss functions are mean squared error (MSE) (see Figure 4), and cross entropy loss.

A special type of activation function called softmax is used on the weighted sum of the weights of the connections to the output layer, to produce an input vector to the output layer, representing the probability distribution of the input sample belonging to each class.

In this example, the input vector to the output layer could look like; $[0.60, 0.40]$. This means that the ANN estimates the likelihood that the input data belongs to class 1 is 60 %, while the likelihood that it would belong to class 2 is 40 %.

The label vector of the output data could either look like $[1,0]$, which means that the true label of the input data is class 1, or look like $[0,1]$ which means that the input belongs to class 2. The format of a vector with one value of 1, and all other values are 0, is called one-hot encoding.

$$MSE := \frac{1}{n}\sum_{t=1}^{n} e_t^2$$

**Figure 4:** The mean squared error (MSE) function. In a multi-class classification problem; *n* is the number of possible classes, *t* is the index for a specific class and, *e* is the predicted value subtracted by the true value.

Each weight in the network is then updated a new value, equal to the gradient of the loss, with respect to its own weight, multiplied by a *learning rate*. The learning rate is a small value between zero and one, which determines how fast the model should learn recognizing patterns in the input data. If the learning rate is too small, the process of training the network will take an unnecessarily long time, but if it is too big, the model might miss the optimal solution.

This process is either repeated a predefined number of times, called *epochs*, or is repeated until a defined acceptable performance level has been reached. Accuracy and loss are two performance metrics. Accuracy is defined as the number of correct predictions, divided by the total number of predictions made.

### Evaluation of an ANN

When training an ANN, all input samples will be split into three distinct sets; a training set, a validation set, and a test set.

Based on the training set, the ANN will try to optimize the weights between its nodes. The validation set is used to evaluate the performance of the ANN under the training process, and see if the model *overfit* or *underfit* the training data, or not. The weights are not updated based on the validation set.

Overfitting means that the model learns the patterns of the training data and becomes good at classifying its samples, but fail to generalize that knowledge to data samples that it has not been trained on. In order to prevent a model from overfitting, the following strategies can be useful.

- A larger training set give the model a more generalized view.
- It is possible that the model are too complex for the input data, and therefore becomes very specialized, but fail to generalize. Therefore, a simplified model might increase the performance of the model.
- A technique called *dropout*, which randomly inactivate nodes under training process, prevents the model from becoming too specialized.

Underfitting means that the model is bad at classifying samples, whether it is from the training or validation set. In order to prevent a model from underfitting, a more complex ANN can be developed, and/or the sample data can be pre-processed with feature extraction (see Section 2.5).

The test set is used to evaluate the performance of the ANN after the training process, and the result is often presented by a confusion matrix (see Section 2.6).

## 2.3 Deep Neural Network

A deep neural network (DNN), is an ANN with more than one hidden layer. In a DNN of several layers, the first hidden layers look for small features, while the hidden layers at the end combine these small features analyze the data in a broader perspective.

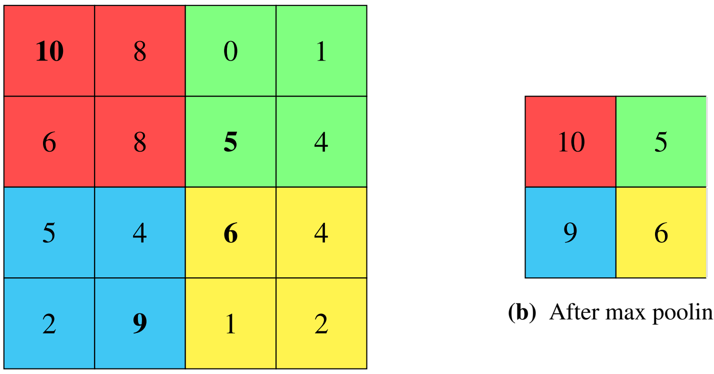
## 2.4 Convolutional Neural Network

A convolutional neural network (CNN) is a special type of ANN:s that are great when the position of the data matters, for example in images and audio.

Whats differentiates a CNN from an ANN, is that it has at least one special type of hidden layer, a *convolutional layer*. The nodes in a convolutional layer only receive input from a portion of the the previous layer, called *receptive field*. Each node in a convolutional layer uses *filters* to detect features within its receptive field of the previous layer. The convolution technique reduces the number of computational operations used by each node.

In a CNN with multiple convolutional layers, the first convolutional layer(s) look for small features, while the last convolutional layer(s) combine these small features and look for bigger features.

A CNN may also have a hidden layer called *pooling layer*, which can be described as a filter that slides over the input data, each step according to a fixed filter size, to combine nodes from the previous layer into one node, to reduce the number of nodes in the next layer [10].



| | | | |
|---|---|---|---|
| **10** | 8 | 0 | 1 |
| 6 | 8 | **5** | 4 |
| 5 | 4 | **6** | 4 |
| 2 | **9** | 1 | 2 |

**(a)** Before max pooling.

| | |
|---|---|
| 10 | 5 |
| 9 | 6 |

**(b)** After max pooling.

**Figure 5:** Figure 5a shows a 4 by 4 feature set before passing through a max pooling layer that shrinks the set by a factor of two per dimension. Figure 5b shows the same feature set after passing through the max pooling layer.

The use of convolutional and pooling layers makes the CNN well suited for computational expensive recognition tasks.

## 2.5 Feature Extraction

One challenge with neural networks is how to handle the size of the original input data, which is often very large in terms of memory. For example, image and audio files often come in the size of several MB. This makes the training process very expensive, both in terms of memory allocation and the number of computational operations needed.

By pre-processing the original input data and extract specific features, and instead use those features as input to the neural network, the size of the input data is reduced and hence also the number of computational operations needed to train the network. Pre-processing of the original data also often makes the performance of the model increase, because redundant information is reduced [11].

A feature that has been extracted from an audio file containing read sentences, could for example represent the pitch (if the sentence is read with a high or low tone) or the volume (if the sentence is read with a powerful voice or not).

### 2.6 Confusion Matrix

A confusion matrix is a table that visualize the performance of a classification model, commonly a model trained under supervised learning. The table present the predictions made on one axis and the actual class on the other. If all predictions are correct, the confusion matrix will have a straight diagonal line, from cell (1, 1) to ($n$, $n$) of predictions, where $n$ are the number of possible emotions, and all other cells are empty.

There are two types of confusion matrices, *normalized* confusion matrices and confusion matrices *without normalization*. A normalized confusion matrix visualizes the distribution of the predictions, whereas one without normalization shows the predictions made in absolute numbers.

# 3 Materials

In this section, all materials used in this study, such as data, software and hardware and are specified.

## 3.1 Data

The audio files of emotional speech, the different types of output data from the feature extraction tool openSMILE, and the sets of emotions used in the study will be presented in this section.

### Emotional Speech Database

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [12] provided all the audio files used in this study. Each audio file is 3 seconds long and contains speech classified as one specific emotion. The format of the audio files are 16bit, 48 kHz WAVE-format (.wav).

The library `Audio_Speech_Actors_01-24.zip` (215 MB), contains one subdirectory for each of 24 actors reading 60 sentences, which contributes to a total of 1440 audio files. Each file name has the format `xx-xx-xx-xx-xx-xx-xx.wav`, where each column `xx` is an identifier that distinguishes the files (see Table 1).

The first and second column specifies the file format and whether the person speaks or sings. Every file is an audio file (`03`) and contains speech (`01`).

The third column specifies one of eight emotional expressions: neutral (`01`), calm (`02`), happy (`03`), sad (`04`), angry (`05`), fearful (`06`), disgust (`07`), and surprised (`08`).

The fourth column specifies the emotional intensity, where (`01`) means normal intensity and (`02`) means strong intensity. For every emotion except normal emotion, both normal and strong intensity are represented, while there is only normal intensity for the normal emotion. Strong intensity means that the actor speaks with a more powerful voice than usual.

The fifth column specifies which statement that is read, which is either ”Kids are talking by the door” (`01`) or ”Dogs are sitting by the door” (`02`).

All sentences are repeated two times. The corresponding repetition is displayed in the sixth column.

There are 12 male and female actors each. Which actor that speaks is specified in the seventh column, where an odd number means male and even means female.

**Table 1** Audio file naming convention specification. The meaning of each two digit number in the audio file `03-01-03-02-01-02-05.wav` are presented in the table.

| Column | Characteristic | Identifier | Meaning |
|---|---|---|---|
| Column 1 | Modality | 03 | Audio-only |
| Column 2 | Vocal channel | 01 | Speech |
| Column 3 | Emotion | 03 | Happy |
| Column 4 | Emotional intensity | 02 | Strong |
| Column 5 | Statement | 01 | ”Kids are talking by the door” |
| Column 6 | Repetition | 02 | 2nd repetition |
| Column 7 | Actor | 05 | Actor number seven (male) |

The database is released under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

RAVDESS was chosen as database for this study based on the following factors;

- The database consists of English sentences only.

  We think that it is a good idea to start training our model on one language, and then evaluate its performance. Comparing two models, one trained with sentences in one language and the other trained on sentences in several languages could yield interesting results, but that is not the focus in this study.

- Each audio file had the same the length.

- The database were well structured and the label of the audio files made it easy to loop through the database and select specific files.

- The database is free to use for academic purposes.

### Feature Sets

With openSMILE, three different pre-processing configurations were used to extract features from the the audio files described in Section 3.1. We are using two of the same sets that Mena [4] is using in his thesis, namely, *Feature set 1* and *Feature set 2*. Furthermore, another set were created to use the strength of a CNN, which is that it can detect features from structured data (think of the data as a spectrogram), which means that feature extraction is not necessary.

**Feature set 1**
The first set of features is based on a modified version of openSMILE's default configuration file `emobase2010.conf`, to output data as a comma separated value (CSV) file. The output consists of 1582 features with 34 low-level descriptors which gives the dimension $1582 \times 1$.

**Feature set 2**
The second set is using openSMILE's default configuration file `IS09_emotion.conf`, which outputs a CSV file with 384 features as statistical functionals applied to low-level descriptor contours and has the dimension $384 \times 1$. Most of the features in this set also exists in feature set 1 as feature set 1 is an improvement of this set.

Since `IS09_emotion.conf` default output is in wrong format, the flags presented in Table 2 had to be used to get the output in CSV format.

**Table 2** Command line flags used when running openSMILE's configuration file `IS09_emotion.conf`.

| Flag | Command |
|---|---|
| -csvoutput path | Makes it output in CSV format to the given path. |
| -timestampcsv 0 | Skip the timestamp column when outputting. |

### Raw data

An audio file does always contain some sort of feature and therefore we are using a feature set consisting of only one feature as our raw data to compare features with raw audio data. Our raw data uses openSMILE's `demo1_energy`, which windows the audio into separate frames of 10 milliseconds each and extracts log energy for each frame. The output is saved in a CSV file. Since each audio file is not exactly the same length, the number of frames differ. Therefore, removing the end of each file to make it 295 frames it could be used in our CNN. Since in every file ends with a silent portion, this should not have a big impact on the outcome of the classification. The single feature extracted exists in the other sets as well.

### Training, test and validation set

The data set is divided into three sets; train, validation, and test. The training set will be used to learn the model to recognize patterns from the input data by adjusting its weights. The validation set will be used to evaluate the performance of the model at the end of each training epoch, to be able to stop the training process if the model start overfit the data. (Notable is that although the technique of early stopping was implemented in the model, it was never used). After the model has finished the training process, the model will be evaluated with the test set.

The data of two female and two male actors are randomly selected from the data set to be used as test data. The validation data is chosen by Keras by randomly selecting 20 % of the training data as validation data and training consist of the remaining 80 %.
Training, test, and validation sets are normalized to be between 0 and 1.

### Emotion Sets

RAVDESS contains audio files representing eight different emotions.

It may be interesting to compare the feature sets on different amount of emotion classes and therefore three sets with eight, five, and three emotion classes were created for each feature set, presented in Table 3.

The emotion set with three emotions intersects the emotion sets of five and eight emotions, and the emotion set of five emotions intersects the emotion set of eight emotions. This makes it easier to compare the result and draw conclusions.

**Table 3** The three different emotion sets used in this study.

| Emotion set 8 | Emotion set 5 | Emotion set 3 |
|---|---|---|
| Neutral | Calm | Angry |
| Calm | Angry | Disgust |
| Happy | Fearful | Surprised |
| Sad | Disgust | |
| Angry | Surprised | |
| Fearful | | |
| Disgust | | |
| Surprised | | |

The emotions within each feature set were chosen randomly.

## 3.2 Software

The programs used in this study will be presented in this section.

### openSMILE

Many different features has been used for emotion and speech recognition, e.g Zheng et al. [5] used preprocessed speech signals with reduced dimension and splitting the signals into non-overlapping segments, and Abdel-Hamid et al. [8] used Fourier-transform-based filter-bank analysis to generate feature. The issue is that many of the features were created using their own implementation. Since we have limited time for this thesis we can not implement our own solutions for feature extraction. Mena [4] used a software called openSMILE to generate features. openSMILE [13] is a software that extracts features from audio files and is used worldwide by researchers and companies in the speech recognition field. With several predefined configurations, it makes it easy for us to choose features and process a batch of audio files in a simple script [14].

### TensorFlow

TensorFlow [15] is an open-source machine learning library developed by Google and can be used to model convolutional neural networks. Since the process of training and testing a CNN is computational intensive, the version of TensorFlow with graphic processing unit (GPU) support was chosen, instead of the one with only central processing unit (CPU) support.

The installation was done with the package management system pip3. Note that TensorFlow requires Python (version 3.5 or higher) 64-bit on Windows.

### Keras

Keras [16] is a high-level application programming interface (API) library written in Python that can work on top of TensorFlow, with the purpose of facilitating the process of building neural network models. The installation of Keras version 2.1.5 was done with the package management system pip3.

The optional dependency Graphviz (stable release version 2.38) is an open source graph visualization software. Pydot, which is a Python interface to Graphviz, were used to plot the CNN model (see Figure 6). Graphviz was downloaded from `https://www.graphviz.org/` and pydot was installed with pip3.

## 3.3 Program Language

The program language and libraries used for the implementation of the tests will be presented in this section.

### Python

The model was implemented in the high-level programming language Python, version 3.6.5 64-bit. Here follows some Python modules that we used for this thesis:

- The `NumPy` [17] library offers user friendliness along with efficiency when handling multi-dimensional arrays.
- The `Pandas` [18] library provides a smart way of managing numerical tables.
- `pickle` is a module that enables the user to save python objects to file and load them for usage later.
- `Matplotlib` [19] is a library that enables the user to make high quality graphical figures for publications.
- The `scikit-learn` [20] library provides tools to make confusion matrices.

## 3.4 Devices

### Operating System and Hardware

All training and testing were done on Windows 10 Pro machines and run on its GPU. For details see Table 4.

**Table 4** Device specification.

| | |
|---|---|
| Processor | Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz |
| RAM | 16 GB |
| GPU | NVIDIA GeForce GTX 1050 Ti |

# 4 Method

When recognizing speech, a convolutional neural network has been shown to produce great results [8]. Other methods, such as HMM and SVM, together with selected features, associated with emotions in speech, had high results [7] classifying emotions in speech. While a DCNN was not able reach the same high accuracy it can still outperform a SVM when using certain preprocessed speech signals [5]. It seems like using different combinations of classification models and features to recognize emotions in speech affects the performance. It becomes interesting to see what impact different feature sets have on different classification models. Mena [4] compared three different classifiers with three different feature sets, we can use a similiar method with a classifier he did not use. In this study, three different feature sets will be compared using a CNN. The accuracy, as well as confusion matrices, for each feature set will be compared to see how feature extraction affects accuracy.

## 4.1 Network Architecture

A convolutional neural network (CNN) model for a multi-class classification problem was developed in Keras, using the Keras Sequential model. Although each different data set of features will probably have its data best classified with its own unique CNN model, the focus of this study is to compare the impact of feature extractions when classifying emotional speech. Therefore, only one CNN model was used in the comparison. Note however, that because the different set of features have different length, the input layer in the CNN adjusts its dimension dynamically. The CNN model is visualized in Figure 6.

The first two hidden layers are convolutional layers of 124 nodes each, and both of them uses the ReLU as activation function. The third hidden layer is a (max) pooling layer, followed by a dropout layer which randomly inactivates 30 % of the nodes to prevent overfitting. The last hidden layer is a fully connected layer, where each node of its nodes are connected to every node in the previous layer. The output layer uses the softmax activation function.

The CNN uses categorical crossentropy as loss function, uses a learning rate of 0.0001, and is trained under 100 epochs.

We designed the network architecture by experimenting and comparing how different hidden layers affected its performance and this architecture yielded the highest accuracy.

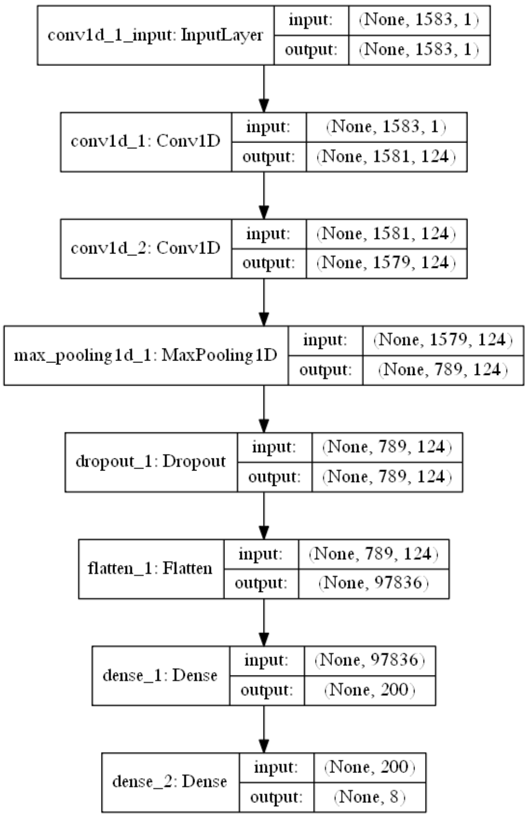


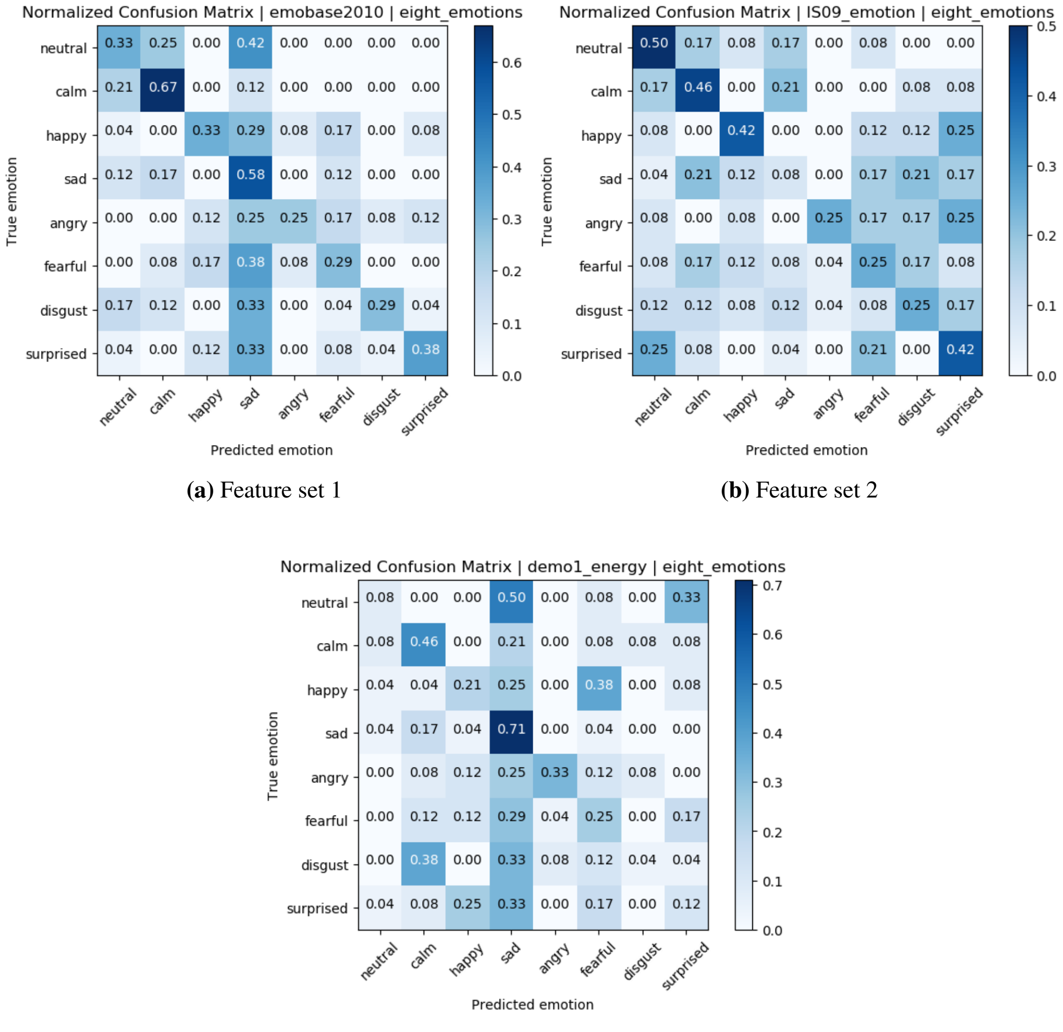
**Figure 6:** A graphic visualization of the CNN design used in an eight emotion multi-class classification problem with emobase2010 as feature extracting configuration.

## 5 Result

The data in Table 5 presents the proportion of correctly made predictions and indicates that the feature set 1 provide our CNN with features that gives highest accuracy, followed by feature set 2. The raw data did not perform as well compared to the others. It becomes easier to distinguish emotions when the number of emotions to classify becomes smaller, as can be seen on the accuracy, which is increasing with fewer classes, in Table 5.

**Table 5** Accuracy for validation sets.

| | eight_emotions | five_emotions | three_emotions |
|---|---|---|---|
| emobase2010 | 0.39 | 0.53 | 0.69 |
| IS09_emotion | 0.32 | 0.38 | 0.58 |
| demo1_energy | 0.29 | 0.38 | 0.69 |



**(a)** Feature set 1

**(b)** Feature set 2

**(c)** Raw data

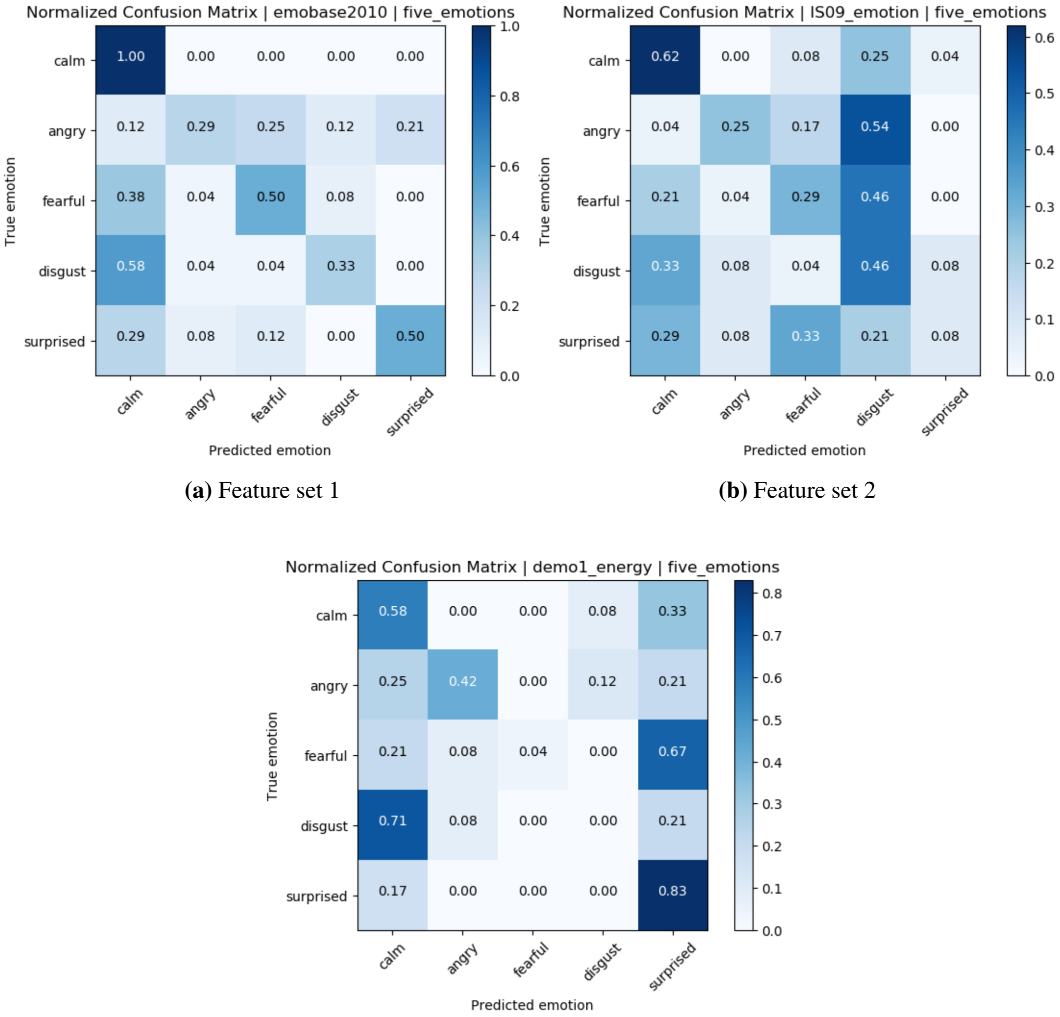
**Figure 7:** Confusion matrices for each feature set with 8 emotions. A training set of 1260 sentences and a test set of 180 sentences were used.

### 5.1 Confusion Matrices of Simulations with Eight Emotions

It is shown in Figure 7 and Table 5 that feature set 1 has the best result, in terms of accuracy, for eight emotions with 39 % accuracy. It becomes better at distinguishing between emotions overall, but at the same time getting more confused by feelings such as neutral and sad which can be seen when looking at Figure 7a.

In Figure 7b, feature set 2 has an accuracy of 32 %. It shows almost the same results as feature set 1 when it comes to distinguish emotions but have a harder time separating emotions from fearful, disgust and surprised, and sad.

The raw data have an accuracy of 29 %. The spread in predictions, when looking at Figure 7c, is not as high as the other but still performs decently overall.



**(a)** Feature set 1

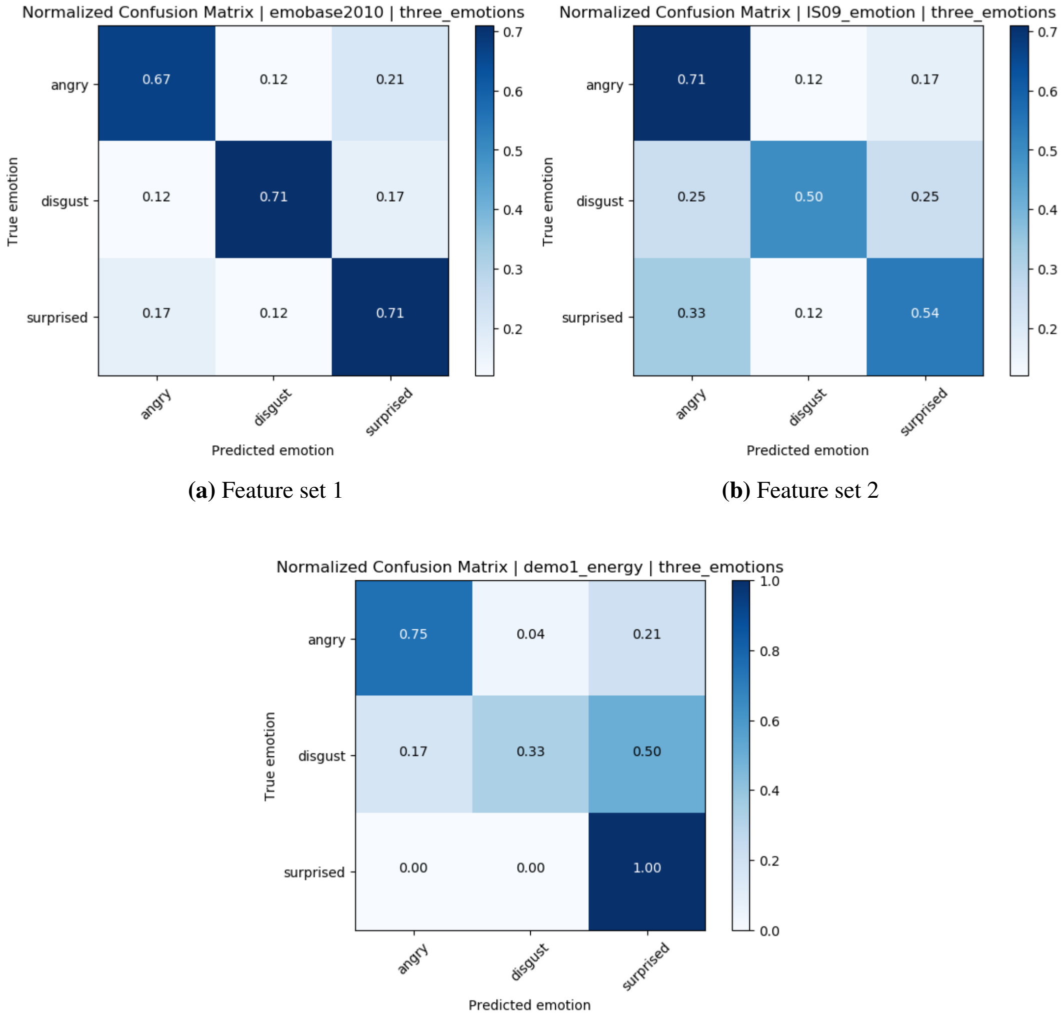
**(b)** Feature set 2

**(c)** Raw data

**Figure 8:** Confusion matrices for each feature set with 5 emotions. A training set of 840 sentences and a test set of 120 sentences were used.

## 5.2 Confusion Matrices of Simulations with Five Emotions

With five emotions, in Figure 8, feature set 1 still has the best performance with an accuracy of 53 % followed by both feature set 2, and raw data at 38 %. When looking at the calm column for each matrix in Figure 8, it becomes clear that each feature set have problem to distinguish between calm and other emotions. While feature set 1 mostly only has problem with calm, feature set 2 have problem with both calm and disgust and raw data is having issues with calm and surprised.



**(a)** Feature set 1

**(b)** Feature set 2

**(c)** Raw data

**Figure 9:** Confusion matrices for each feature set with 3 emotions. A training set of 504 sentences and a test set of 72 sentences were used.

### 5.3 Confusion Matrices of Simulations with Three Emotions

In Figure 9a feature set 1 has a good overall spread with three emotions and an accuracy of 69 %. Around 70 % on all three emotions.
feature set 2 does not have the same spread or accuracy as feature set 1, as can be seen in Figure 9b but still performs reasonably well with an accuracy of 38 %, having 71 % correct on angry, 54 % on surprised and 50 % on disgust.
In Figure 9c raw data has 38 % accuracy and is really good at finding surprised with 100 % accuracy but is having a hard time distinguishing disgust from surprised with 50 % of disgust is guessed as surprised instead. It manages to find angry with an accuracy of 75 % while mistaking it as surprised 21 % of the times.

## 6 Discussion

With eight and five possible emotions, the CNN gave better predictions with feature extraction, instead of raw data (see Table 5). When comparing our accuracy to Zheng et al. [5] results, in which they got 40 % accuracy, we get higher accuracy with feature set 1 and almost the same with the other feature sets. Since they used almost the same classification model we argue that using feature set 1 yields higher accuracy than PCA whitened spectogram segments. They also used five emotions to classify and comparing their five emotions with our set of five emotions we got an increase in accuracy by 13 % with feature set 1. Lin and Wei [7] used a few selected features and managed to reach above 88 % with SVM and HMM which indicates that more features does not necessarily improve accuracy but rather selecting certain features the accuracy can be increase at the reduced cost of training resources. This concurs with the results from Mena [4] that when he removed features the accuracy was either the same or higher. For a CNN it seems that a higher number of features improves the accuracy when classifying eight or five emotions. With three possible emotions both feature set 1 and raw data produced the same accuracy while feature set 2 can not keep up. The most likely reason as to why they still have very close results with three emotions, is that the data used is not enough to properly separate emotions from each other since many of the audio files used sound very similar even though they are different emotions, which in turn makes many of the features not matter as much. This corresponds to the results from Mena's [4] study where they showed that fewer features produce almost the same results.

With the raw data, we expected the results to be higher than the other using a CNN. Instead, the raw data produced the lowest accuracy in eight and five emotions. A CNN is dependent on the position of the features and can come to conclusions using that, but with only one feature being available the positions of the raw data frames did not matter too much. Feature set 1 and feature set 2 does not have that dependency with their features which makes the sets not suitable for a CNN and when comparing the results with Mena [4], and Lin and Wei [7], this becomes clear as the accuracy of our CNN was nowhere near the accuracy of their results. But both feature set 1 and feature set 2 still outperformed the raw data even though they are statistical features. This implies that features actually do matter, even for a CNN, and since raw data is not far behind in accuracy it can be interesting too see how well the feature sets would perform when used as features over time, rather than statistical functions.

Something noteworthy is that some emotions are often mistaken as other emotions and the pattern in the confusion matrices indicates that these emotions sounds very similar to each other. For example, in Figure 8b disgust and calm have 192 % respective 149 % and it seems as calm and disgust is hard to distinguish from other emotions. Emotions can be expressed very different for each individual and can sound the same when an individual express different emotions. It is therefore hard to distinguish emotions between only few individuals. With more data we believe that the network could become better at distinguishing actors different emotions since the small difference may be clearer with a larger set of data.

We think that the volume of data is nowhere near enough to be able to train a network to the level that it can provide high enough accuracy to be useful in real life situations. However, the predictions could be used as a complement, together with other factors, such as sentiment, facial expression and body posture analysis, similar to what Kahou et al. [3] did in their study. We argue that the use of a CNN is useful in emotions in speech classification problems with a different set of features, but a CNN will not outperform other models such as SVM or HMM with statistical features [7]. A CNN can be used as a simple method to easily implement a classification model that can decently classify a few set of emotions in speech.

## 7 Conclusion

We studied the effect from feature extraction on audio using a convolution neural network to classify emotions. We experimented with three sets of features (feature set 1, feature set 2, and raw data) on eight, five and three classes for each feature set to measure accuracy when classifying emotions. Results show that feature set 1 outperformed the other feature sets with eight, five and three emotions to classify with an accuracy of respective 39 %, 53 % and 69 % which got a better result than using principal component analysis as Zheng et al. [5] did. We did not manage to reach very high accuracy due to the low amount of data used and is nowhere near the results as Mena [4]. Many of the features may be redundant and can be removed without affecting the accuracy of our model. A CNN may not be optimal to use in emotion recognition but, it can prove to be useful for simple applications since pre-processing data is not necessary.

## 8 Suggestions for Further Work

Since the features we used were not optimized for a CNN it would be interesting to try other features that are more suited to a CNN and see how well a CNN actually can perform. Earlier research have had high accuracy recognizing speech, but very few have done well in recognizing emotions in speech. A reason could be the features used for speech recognition may not be useful for recognizing emotions in speech, which Lin and Wei [7] also stated. The low amount of data could be a reason why the accuracy is quite low. In further research, we believe having a larger dataset might provide higher accuracy.

The data used were only from English sentences and one may ask what happens when the CNN is used to predict other languages. By testing this one can decide whether or not a training set containing different languages is needed or not.

We suggest that further research in any of the topics mentioned above is relevant and interesting to further improve classification of emotions.

## References

[1] Arsénio Reis, Hugo Paredes, Isabel Barroso, Maria João Monteiro, Vitor Rodrigues, Salik Ram Khanal, and João Barroso. Autonomous Systems to Support Social Activity of Elderly Eeople - A Prospective Approach to a System Design. In *2016 1st International Conference on Technology and Innovation in Sports, Health and Wellbeing (TISHW)*, pages 1–5, Dec 2016.

[2] Arsénio Reis, Dennis Paulino, Hugo Paredes, and João Barroso. Using Intelligent Personal Assistants to Strengthen the Elderlies' Social Bonds - A Preliminary Evaluation of Amazon Alexa, Google Assistant, Microsoft Cortana, and Apple Siri. In *International Conference on Universal Access in Human-Computer Interaction*, pages 593–602. Springer, 2017.

[3] Samira Ebrahimi Kahou, Christopher Pal, Xavier Bouthillier, Pierre Froumenty, Çaglar Gülçehre, Roland Memisevic, Pascal Vincent, Aaron Courville, Yoshua Bengio, Raul Chandias Ferrari, et al. Combining Modality Specific Deep Neural Networks for Emotion Recognition in Video. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, pages 543–550. ACM, 2013.

[4] Marc Escalona Mena. Emotion Recognition from Speech Signals. 2012. Faculty of Electrical Engineering at University of Ljubljana.

[5] W.Q. Zheng, J.S. Yu, and Y.X. Zou. An Experimental Study of Speech Emotion Recognition Based on Deep Convolutional Neural Networks. In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 827–831. IEEE, 2015.

[6] Olle Sundin. Impact of Noise Reduction when Classifying Bird Song with CNN. 2017. Department of Computing Science at Umeå University.

[7] Yi-Lin Lin and Gang Wei. Speech Emotion Recognition Based on HMM and SVM. In *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, volume 8, pages 4898–4901. IEEE, 2005.

[8] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional Neural Networks for Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(10):1533–1545, 2014.

[9] deeplizard. Artificial Neural Networks explained. `https://www.youtube.com/watch?v=hfK_dvC-avg`. [Online; accessed 2018-05-28].

[10] deeplizard. Max Pooling in Convolutional Neural Networks explained. `https://www.youtube.com/watch?v=ZjM_XQa5s6s`. [Online; accessed 2018-05-28].

[11] Pádraig Cunningham. Dimension Reduction. In *Machine Learning Techniques for Multimedia*, pages 91–112. Springer, 2008.

[12] Steven R. Livingstone and Frank A. Russo. The Ryerson Audio-Visual Database of Emotional Epeech and Song (RAVDESS): A Dynamic, Multimodal Set of Facial and Vocal Expressions in North American English. *PloS one*, 13(5):e0196391, 2018.

[13] open-Source Media Interpretation by Large Feature-Space Extraction. `https://www.audeering.com/research-and-open-source/files/openSMILE-book-latest.pdf`. [Online; accessed 31-July-2018].

[14] Florian Eyben, Felix Weninger, Florian Gross, and Björn Schuller. Recent Developments in openSmile, the Munich Open-Source Multimedia Feature Extractor. In *Proceedings of the 21st ACM International Conference on Multimedia*, pages 835–838. ACM, 2013.

[15] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. Software available from tensorflow.org.

[16] François Chollet et al. Keras. `https://keras.io`, 2015.

[17] Travis E. Oliphant. A Guide to NumPy. 2006. USA: Trelgol Publishing.

[18] Wes McKinney. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, 2010.

[19] John D. Hunter. Matplotlib: A 2D Graphics Environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[20] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.