Tampere University

Johannes Simulainen

AUTOMATIC DETECTION OF ENTHUSIASM FROM SPEECH USING ACOUSTIC FEATURES

Bachelor's thesis
Faculty of Information Technology and Communication Sciences
Examiner: M. Sc. Einari Vaaras
May 2022

ABSTRACT

Johannes Simulainen: Automatic Detection of Enthusiasm from Speech Using Acoustic Features
Bachelor's thesis
Tampere University
Computing Sciences
May 2022

In audio processing, speech emotion recognition (SER) is concerned with automatic recognition and extraction of emotion from speech data. One subcategory of SER is the automatic recognition of enthusiasm. The automatic recognition of enthusiasm could benefit virtual agents and robots, especially in teaching and communication.

Recently, a multimodal dataset focusing on enthusiasm, Entheos, was released, along with baseline models for detecting enthusiasm. In the present study, different classifiers for detecting enthusiasm were examined and compared to Entheos' baseline model.

The experiments showed that the multilayer perceptron was the best at enthusiasm recognition, while the convolutional neural network was the best at distinguishing different levels of enthusiasm. Using acoustic-only features, our tested models outperformed the baseline model regardless of the features used by the baseline network, even when the baseline network used multimodal features. We observed that the biggest improvement in performance compared to the baseline model was achieved with the standardization of the features, although model choice and architecture also had an impact. Potential problems related to the wider applicability of the results are also discussed.

Keywords: enthusiasm detection, speech emotion recognition, speech processing, eGeMAPS features, acoustic features

TIIVISTELMÄ

Johannes Simulainen: Innokkuuden automaattinen tunnistaminen puheesta akustisia piirteitä käyttäen
Kandidaatintyö
Tampereen yliopisto
Tietotekniikka
Toukokuu 2022

Puheen tunteiden tunnistuksessa (SER; speech emotion recognition) tavoitteena on automaattisesti tunnistaa ja luokitella tunteita puheesta. Eräs SER:in alakategoria on innokkuuden automaattinen tunnistus, jota hyödyntämällä voitaisiin esim. parantaa ihmisten vuorovaikutusta virtuaaliagenttien sekä robottien kanssa, etenkin opetuksessa ja kommunikaatiossa.

Hiljattain julkaistiin innokkuuteen keskittyvä multimodaalinen tietoaineisto nimeltään Entheos, jonka mukana julkaistiin myös ns. ”baseline”-luokitinmallit perustulosten saavuttamiseksi kyseisellä tietoaineistolla. Tässä tutkimuksessa kokeiltiin erilaisia luokitinmalleja innokkuuden tunnistamiseen, sekä vertailtiin näitä luokittimia Entheoksen perustulosten malleihin.

Työssä verratuista luokittimista monikerroksinen perseptroniverkko (engl. multilayer perceptron) oli tarkin innokkuuden tunnistamisessa, kun taas konvoluutioneuroverkko (engl. convolutional neural network) erotti parhaiten innokkuuden eri tasoja. Kaikki työssä kokeillut luokittimet suoriutuvat Entheoksen perustulosten saavuttaneita malleja paremmin pelkkiä akustisia piirteitä käyttäen, vaikka perustason mallilla oli käytössä piirteitä useammasta eri modaliteetista. Tutkimuksen perusteella eniten tuloksia parantanut tekijä oli akustisten piirteiden standardointi, mutta luokitinarkkitehtuureilla oli myös jonkin verran vaikutusta tuloksiin. Työn lopussa käsitellään myös tulosten laajempaan soveltuvuuteen liittyviä mahdollisia ongelmia.

Avainsanat: innokkuuden tunnistus, puheen tunteiden tunnistus, puheenkäsittely, eGeMAPS-piirteet, akustiset piirteet

PREFACE

This thesis, being my first step into academic work, is the first major milestone in my journey to a master's degree. The biggest thanks go to the examiner of this thesis, M. Sc. Einari Vaaras, for providing me with this interesting topic, and moreover, for his excellent guidance and help during the project. I would also like to express my gratitude to the staff at the University of Tampere for teaching me about a multitude of interesting subjects, some related to this thesis. I am also grateful to my friends and family for creating the best possible environment for working on the thesis and my studies in general.

Tampere, 10th May 2022

Johannes Simulainen

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CNN	Convolutional neural network
eGeMAPS	Extended Geneva Minimalistic Acoustic Parameter Set
GeMAPS	Geneva Minimalistic Acoustic Parameter Set
LLD	Low-level descriptor
MLP	Multilayer perceptron
PSP	Paralinguistic speech processing
RBF	Radial basis function
ReLU	Rectified linear unit
SER	Speech emotion recognition
SVM	Support vector machine

1. INTRODUCTION

Speech contains information in a multitude of ways. Maybe the most obvious is the linguistic content of speech, which consists of e.g. phonetics, grammar and semantics of the language (Schuller and Batliner, 2013, p. 3). However, there is much more information in speech than just the linguistic content. For example, a sentence "My day was amazing" can be interpreted differently depending on how it was said. This sentence could be interpreted truthfully if it were pronounced enthusiastically, or vice versa if it were pronounced monotonously. You could even infer the health of the speaker due to e.g. coughing. The information that leads to these interpretations could be said to be alongside linguistics, or "paralinguistic" (Schuller and Batliner, 2013, p. 3). As defined by Schuller and Batliner (2013), paralinguistics is concerned with *how* something was said instead of *what* was said. When the analysis of the how is done by, or with the help of computers, it is called computational paralinguistics, also known as paralinguistic speech processing (PSP).

As demonstrated in the previous paragraph, the emotional state of the speaker is a part of paralinguistics. A subcategory of PSP concerned with recognizing and extracting emotions automatically is called speech emotion recognition (SER). The automatic recognition of emotions would make human-machine interactions more natural. Thus, SER has uses in in-car board systems, call center applications and even diagnostic tools for therapists (El Ayadi et al., 2011). There are numerous challenges associated with SER that slow down its application in practice. These include e.g. how emotions appear differently in different cultures and how a single sentence can have multiple emotions (El Ayadi et al., 2011).

According to Viegas and Alikhani (2021), one important emotion that has gotten relatively small amount of attention in SER is enthusiasm, despite it being important in teaching and communication in general. Better recognition of enthusiasm and more enthusiastic behaviour could benefit e.g. virtual agents and robots (Viegas and Alikhani, 2021).

In this thesis, we compare different classifiers for enthusiastic speech detection against a baseline model introduced in a recently published enthusiastic speech dataset, Entheos (Viegas and Alikhani, 2021), using already extracted acoustic features. In Chapter 2, a typical classification pipeline for PSP (an thus SER) is introduced. Chapter 2 also covers recent developments in the detection of enthusiasm in speech processing. Chapter 3 contains information about the acoustic feature set and the classifiers used in the present study. The dataset used and the experimental setup are described in Chapter 4. The results of the experiments are presented in Chapter 5, while the conclusions based on the results are drawn in Chapter 6.

2. BACKGROUND

Although PSP systems differ depending on the details of the task, there are certain important building blocks that are commonly used. Generally, a PSP system includes feature extraction, where relevant features get extracted from labeled speech data, and a classifier, which is trained to classify speech using the extracted features. Afterwards, the classifier is used to predict the label of unlabeled samples (Schuller and Batliner, 2013, pp. 179–184). In the vast majority of cases learning is supervised, necessitating labeled data, but there have been some recent developments in unsupervised speech recognition systems (Baevski et al., 2021).

As mentioned previously, feature extraction is an important part of a PSP classification pipeline. By using a smooth windowing function, the audio signal is divided into 10–30-ms windows, also known as audio frames, from which short-time features are extracted from. Typically, there is overlap between these windows to avoid information loss within the signal. Commonly extracted short-time features include intensity, intonation, Mel frequency cepstral coefficients and linear prediction cepstral coefficients (Schuller and Batliner, 2013, pp. 179–189; Wani et al., 2021). These short-time features are sometimes referred to as low-level descriptors, or LLDs.

Paralinguistic phenomena, such as the emotional state of a person, are rarely contained to a single frame (Schuller and Batliner, 2013, pp. 179–183). Therefore, it is necessary to examine how the short-time features develop over time. The development of short-time features is examined with supra-segmental features, which are collected from a longer timescale than a singular frame by utilizing the frame-level features. The segmentation is done across varying time scales depending on the application, e.g. SER is mostly concerned with sentences (Schuller and Batliner, 2013, pp. 230–234), while Alzheimer's detection is done across multiple years (Haider et al., 2020). In addition to varying time scales, the length of the segments themselves can vary. From the segments, the frame-level features are extracted as described in the previous paragraph, after which functionals are applied to the time series of the extracted features. Functionals map a series of values of arbitrary length to a single value, creating a single fixed-sized feature vector. A fixed-sized input is a requirement in several classification models, such as decision trees and support vector machines (SVMs), and is therefore convenient for PSP where the length of the input audio samples might vary. Some commonly used functionals include means, moments (e.g., skewness, kurtosis) and percentiles (Schuller and Batliner, 2013, pp. 230–234).

Schuller and Batliner (2013) roughly divide PSP classifiers into static and dynamic classifiers. They differentiate these two categories by their ability to handle differing lengths of feature vectors: static classifiers handle feature vectors of fixed size, while dynamic classifiers can handle feature vectors of varying lengths. Examples of static classifiers include decision trees and SVMs. Dynamic classifiers include hidden Markov models and some artificial neural network architectures, like recurrent neural networks (Schuller and Batliner, 2013, pp. 235–280).

In a more recent literature review, traditional classifiers, like the previously mentioned SVMs and decision trees, were compared with deep learning classifiers, e.g. convolutional neural networks (Section 3.2.3), in the case of SER. According to the review, deep learning classifiers generally outperform traditional methods. Because of this, deep learning classifiers have become more widely used over time. However, traditional methods are still extremely relevant in the field as they require less data for training (Wani et al., 2021).

As mentioned in Chapter 1, enthusiasm is an important part of communication, especially in teaching. In spite of this, research on enthusiasm in a computational context has been limited. Most of the work on automatic detection of enthusiasm has been done in the text domain (Viegas and Alikhani, 2021). Inaba et al. (2011) proposed a method for automatically detecting enthusiasm in text-based utterances using conditional random fields and compared it to an SVM. In the study, the proposed method outperformed the SVM. Lexical feature of the utterance, length of the utterance, lexical cohesion between utterances and the previous output tag were used as the features in the experiments. Tokuhisa and Terashima (2006) analyzed the relationship between utterances and enthusiasm in conversational dialogue. It was found that affective and cooperative utterances are frequent in enthusiastic dialogue.

There are also multiple text-based datasets with labels for enthusiasm or a closely related emotion, such as excitement. Crowdflower (2018), a dataset constructed from 40 000 tweets, has a label called “enthusiasm”. GoEmotions (2020), a dataset constructed from 58 000 curated comments extracted from Reddit, has a label called “excitement” which is described as “Feeling of great enthusiasm and eagerness”.

In SER, enthusiasm has been studied less extensively compared to the text domain. In a review of 37 different emotional speech corpora, enthusiasm didn’t appear as a distinct label. Excitement appeared in four of the reviewed corpora in addition to the one introduced in the paper (Kasuriya et al., 2018). Daido et al. (2014) introduced an aspect of evaluating singing voice called “singing enthusiasm”. An evaluation experiment revealed three acoustic features of voices, which were significantly correlated to the singing enthusiasm values. Two types of regression, linear and logistic, were examined in automatic estimation of singing enthusiasm utilizing the correlated features.

More recently, a multimodal dataset focused on enthusiasm, Entheos (Viegas and Alikhani, 2021), was introduced. Entheos contains 1126 utterances from 113 different TED talk speeches. Different vocal attributes, such as variation and intensity, were evaluated in addition to more intuitive labels, like enthusiasm and emphasis. Regarding enthusiasm, the samples were categorized either as monotonous, normal, or enthusiastic. As a multimodal dataset, Entheos consists of features obtained from audio, video, and transcripts of the utterances. The paper also has a baseline model, which was used in classification with different feature sets. The baseline model performed best with a combination of text-based features and auditory eGeMAPS features (Viegas and Alikhani, 2021). The eGeMAPS features and the Entheos dataset are described in further detail in Sections 3.1 and 4.1, respectively.

3. METHODS

In this chapter two major components of the following pipeline (Figure 3.1) are described.

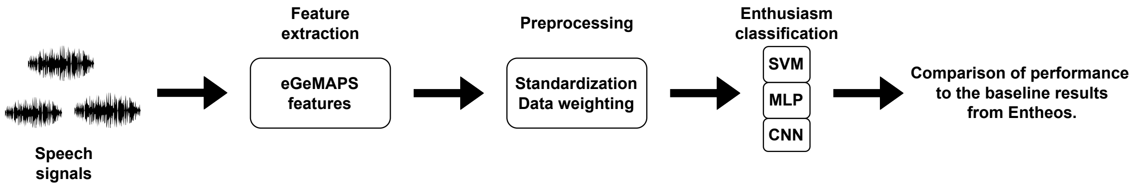

Feature extraction
eGeMAPS features
Preprocessing
Standardization Data weighting
Enthusiasm classification
SVM
MLP
CNN
Comparison of performance to the baseline results from Entheos.
Speech signals

Figure 3.1. *An overview of the pipeline of the present study.*

Section 3.1 describes the feature set that was used in the present study, which is the first part of the block diagram of Figure 3.1 (feature extraction). Section 3.2 describes the "Enthusiasm classification" part of Figure 3.1, where classifiers that were compared to the baseline neural network of Viegas and Alikhani (2021), are introduced.

3.1 eGeMAPS features

As mentioned in Chapter 2, feature extraction is an important part of a PSP pipeline. Thus, the right choice of features improves the classification performance of the system (Swain et al., 2018). In this thesis, an extended version of the Geneva minimalistic acoustic parameter set (GeMAPS), called eGeMAPS, is used, due to it being the best performing acoustic feature set in the experiments conducted by Viegas and Alikhani (2021).

Introduced in 2016, GeMAPS and it's extended version eGeMAPS, are a collection of recommended acoustic parameters for affective vocalizations. The feature set is minimalist in nature to capture the most vital features for general application (Eyben et al., 2016).

Most widely used feature sets, such as the INTERSPEECH challenge sets, have hundreds or even thousands of different acoustic parameters (Eyben et al., 2016). These so called brute-force feature sets have a multitude of problems associated with them that GeMAPS and eGeMAPS features aim to alleviate. One of the problems in large feature sets is the lack of standardization as some of the parameters might be computed differently. This makes comparison of results more difficult. Brute-force sets often overlap only partially, which complicates comparisons even further. GeMAPS and eGeMAPS provide a standardized set of parameters, easing the previously mentioned problems (Eyben et al., 2016).

Additionally, some applications of emotion and mental state recognition require a deeper understanding of the underlying mechanisms, which are unclear when dealing with large number of parameters (Eyben et al., 2016). Reduction in the number of parameters makes interpretation of these mechanisms easier. Reduced parameters also help with over-adaptation of classifiers, which large feature sets are know to promote (Eyben et al., 2016).

GeMAPS' and eGeMAPS' parameters were chosen in accordance with the three following criteria:

1. The potential of an acoustic parameter to index physiological changes in voice production during affective processes.
2. The frequency and success with which the parameter has been used in the past literature.
3. The theoretical significance of the parameter.

Table 3.1 shows some of LLDs chosen by using the previously listed criteria:

Table 3.1. *GeMAPS and eGeMAPS features divided into parameter groups. Frequency and Spectral-groups have additional parameters in the extended set marked with * (Eyben et al., 2016).*

Parameter group	**Example parameter**	**Example parameter description**
Frequency	Formant 1	Bandwidth of first formant
Energy/Amplitude	Loudness	Estimate of perceived signal intensity from an auditory spectrum
Spectral	Alpha Ratio (50–1 kHz, 0.5–1.5 kHz)	Ratio of summed energy
Temporal	Pseudo syllable rate	Number of continuous voiced regions per second
Frequency*	Formant 2–3	Bandwidth of formats 2 and 3
Spectral*	MFCC 1–4	Mel-Frequency Cepstral Coefficients 1-4

After extraction, the chosen LLDs were smoothed with a 3-frame-long symmetric moving average filter. Additional parameters were determined by applying a number of functionals to the LLDs. For example, arithmetic mean and coefficient of variation were applied to all LLDs, and different percentiles were applied to loudness (energy) and pitch (frequency) as functionals. In addition to the parameters found in the minimalistic set, eGeMAPS contains cepstral and dynamic LLDs which are assigned into spectral and frequency parameter groups. Different functionals, such as arithmetic mean and the coefficient of variation, were applied to the supplementary LLDs (Eyben et al., 2016).

The previous acoustic parameters define a fixed size feature vector composed of 62 different features in the minimalistic set and 88 features in the extended set. Despite their relatively small parameter size, GeMAPS and eGeMAPS perform competitively against the much larger brute-forced sets. In their tests Eyben et al. (2016) show that eGeMAPS performs only slightly worse than the ComParE feature set consisting of over 6000 parameters.

3.2 Classifiers

In this section, an overview of the classifiers used in this thesis is given, which represents the third stage in Figure 3.1. The implementation details are given in Chapter 4.

3.2.1 Support vector machine (SVM)

A support vector machine (SVM) is a binary linear classifier (Boser et al., 1992). SVM separates the two classes with an optimal separating hyperplane, which maximizes the margin of separation (Abe, 2010, pp. 1–31). It is frequently used in computational paralinguistics due to it being able to handle large feature spaces and it's robustness to overfitting (Schuller and Batliner, 2013, pp. 242–247).

Let there be M m-dimensional inputs

$$\mathbf{x}_i, \quad i = (1, \ldots, M) \tag{3.1}$$

that belong to class 1 or 2, with associated labels $y_i = 1$ and $y_i = -1$, respectively. If the inputs are linearly separable we can determine the decision function (hyperplane):

$$D(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b, \tag{3.2}$$

where $\mathbf{w}$ is an m-dimensional vector, and b is a bias term. A decision function classifies an input into either class 1 if $D(\mathbf{x}) > 0$, or class 2 if $D(\mathbf{x}) < 0$. This can be written in the form

$$\mathbf{w}^T\mathbf{x_i} + b \begin{cases} > 0 & \text{for } y_i = 1 \\ < 0 & \text{for } y_i = -1 \end{cases} . \tag{3.3}$$

Due to the input data being linearly separable, no data point can satisfy $\mathbf{w}^T\mathbf{x} + b = 0$.

Instead the following inequalities are considered:

$$\mathbf{w}^T\mathbf{x_i} + b \begin{cases} \geq 1 & \text{for } y_i = 1, \\ \leq -1 & \text{for } y_i = -1. \end{cases}, \tag{3.4}$$

which can be rewritten as

$$y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 \quad \text{for } i = 1, \ldots, M. \tag{3.5}$$

There are an infinite number of separating hyperplanes that satisfy this equation. The optimal separating hyperplane has the largest possible margin and can be found by solving the optimization problem

$$\begin{aligned} &\text{minimize } Q(\mathbf{w}, b) = \frac{1}{2}\left\|\mathbf{w}^2\right\| \\ &\text{subject to } y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 \quad \text{for } i = 1, \ldots, M \end{aligned} . \tag{3.6}$$

In the previous optimization problem, it was assumed that the input data is linearly separable, which isn't always the case. Thus, a nonnegative slack variable ξ is introduced. This allows for a feasible solution to exist when the data isn't linearly separable. The optimization problem becomes:

$$\begin{aligned} &\text{minimize } Q(\mathbf{w}, b, \boldsymbol{\xi}) = \frac{1}{2}\left\|\mathbf{w}^2\right\| + \frac{C}{p}\sum_{i=1}^{M}\xi_i^p \\ &\text{subject to } y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \text{for } i = 1, \ldots, M \end{aligned} , \tag{3.7}$$

where $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_M)$ and C is the margin parameter determining the trade-off between margin and misclassification (Abe, 2010, pp. 1–31). The value p is either 1 or 2, with the latter imposing a larger loss for points violating the margin (Tang, 2013).

An SVM can be transformed to classify non-linear tasks if the classes can't be separated well with an optimal hyperplane. This is done by utilizing a kernel trick, where the data points are mapped to a higher dimensional feature space by using a kernel transformation. After being mapped to a higher dimension the classes can be separated with a hyperplane.

As can be seen in Figure 3.2, the crosses can't be linearly separated from the pluses in a 1-dimensional space. After mapping the data to a 2-dimensional space with a quadratic kernel, linear separation is possible without classification errors (Schuller and Batliner, 2013, pp. 242–247). Examples of widely used kernels include the polynomial and radial basis function (RBF) kernels (Abe, 2010, pp. 33–56).

SVMs are formulated for two class problems, which makes extending them to multi-class

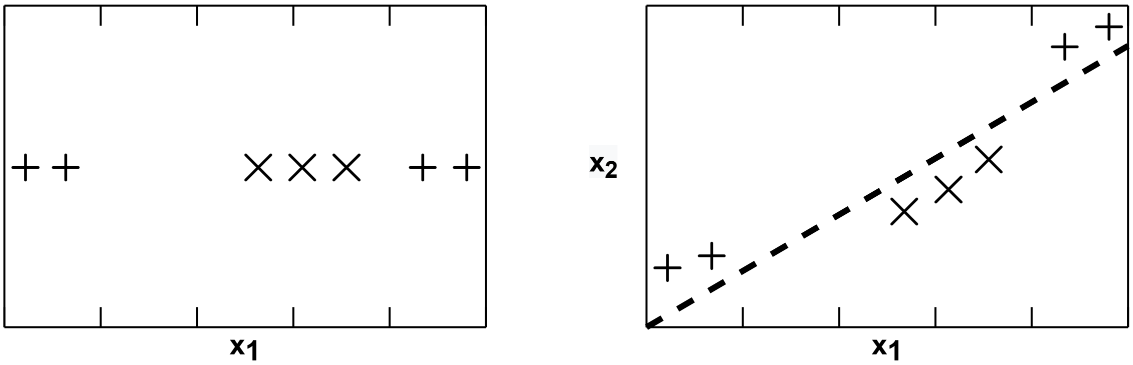


Figure 3.2. An example of kernel trick being used to map non-linearly separable data (left) into a higher dimension. After mapping (right) the data can be separated by the dashed line. Based on a figure from Schuller and Batliner (2013, p. 245).

problems difficult (Abe, 2010, pp. 127–144). Usually multiple two-class SVMs are combined to a single classifier in either one-versus-rest or one-versus-one manner. In one-versus-rest, a binary SVM treats data from a given class as positive and every other class as negative. In one-versus-one, a binary SVM is trained for every pair of classes (Murphy, 2012, pp. 503–504). For example a multi-class classification problem with four classes ('Class 1', 'Class 2', 'Class 3' and 'Class 4') would be divided as follows:

- **Binary SVM 1:** Class 1 vs. Class 2
- **Binary SVM 2:** Class 1 vs. Class 3
- **Binary SVM 3:** Class 1 vs. Class 4
- **Binary SVM 4:** Class 2 vs. Class 3
- **Binary SVM 5:** Class 2 vs. Class 4
- **Binary SVM 6:** Class 3 vs. Class 4

It can be shown that the number of SVMs needed for a multi-class task is $\frac{n(n-1)}{2}$, where n is the number of support vector machines needed when combining them in a one-versus-one manner (Abe, 2010, pp. 127–144).

Model selection is an important aspect of SVMs, as they have multiple parameters that have to be tuned for optimal performance. Selection of kernels is very important for specific applications as it can improve generalization performance (Abe, 2010, p. 33). Kernel parameters, such as degree for polynomial kernel or γ for RBF kernel, have to be optimized based on the chosen kernel. Additionally, the margin parameter C and the slack variable ξ, can be adjusted to control the misclassification error (Abe, 2010, pp. 28–31, 58–60, 93).

3.2.2 Multilayer perceptron (MLP)

A multilayer perceptron (MLP) (Rosenblatt, 1962) is a deep neural network, which tries to approximate some function $\mathbf{f}$ (Goodfellow et al., 2016, p. 168). MLP consists of multiple units called perceptrons, which process weighted inputs and biases, generating an output according to a transfer function (Du, 2014, pp. 5–6). The transfer function, also know as the activation function, is usually used to introduce nonlinearity into the system to make nonlinear classification possible (Goodfellow et al., 2016, pp. 168–177).

In modern neural networks, the rectified linear unit (ReLU) is the most popular activation function (Nwankpa et al., 2018). An output mapped with ReLU stays close to linear despite being nonlinear, which makes the output easier to optimize with gradient based methods. The only difference between ReLU and a linear activation function is that ReLU outputs 0 if the input is negative. Thus, the gradients through ReLU stay large and consistent with no second-order effects that have a negative effect on learning (Goodfellow et al., 2016, pp. 168–195).

Perceptrons are usually grouped into layers, whose width is defined by the number of perceptrons it has. Multiple layers connected together form an MLP. The first layer of an MLP is called the input layer and the final layer is called the output layer. The layers in between of the input and output layers are called hidden layers, due to their values not being given in the data (Goodfellow et al., 2016, p. 6). An input starts at the input layer and flows through the network to the output layer in one direction, making MLP a feedforward network. The appropriate width and depth of the network depends on the application (Goodfellow et al., 2016, pp. 168–177).

During training, depending on the output given by the network, the internal parameters of the perceptrons get altered using an optimization function. The optimization function adjusts the parameters by minimizing the value of a chosen loss function. An example of an optimization function is gradient descent, where small steps are taken towards the local minimum of the loss function (Goodfellow et al., 2016, pp. 82–86, 177–181).

Using gradient descent, a new point is proposed by:

$$\mathbf{x}' = \mathbf{x} - \epsilon \nabla_x f(\mathbf{x}) \tag{3.8}$$

where $\mathbf{x}$ is the original point, $f(\mathbf{x})$ is the function being optimized and ϵ is the learning rate, which is a positive scalar that determines the size of the step (Goodfellow et al., 2016, pp. 82–86, 177–181). The choice of learning rate has a major effect on optimization. If it is set too low, the learning proceeds slowly and might become stuck too early with a high loss value. Too high of a learning rate leads to strongly oscillating values of the loss function (Goodfellow et al., 2016, pp. 294–296).

One central challenge in machine learning is called overfitting. Overfitting occurs when the model conforms too heavily to the training set, creating a wide gap between training error and test error (Goodfellow et al., 2016, pp. 110–119). A way to prevent a neural network from overfitting is dropout. In dropout, random non-output units are removed from chosen layers during training. This creates an ensemble of networks that get combined into one at test time. Model combination nearly always improves the performance of machine learning methods (Srivastava et al., 2014). Dropout is extremely computationally efficient (Goodfellow et al., 2016, p. 265).

3.2.3 Convolutional neural network (CNN)

Convolutional neural network (CNN) (Lecun et al., 1998) is a neural network where at least one layer uses convolution in place of matrix multiplication (Goodfellow et al., 2016, p. 330). CNNs were designed to extract visual features, such as edges and endpoints, from 2D data (Kiranyaz et al., 2021; Lecun et al., 1998). In subsequent layers, detected features are combined based on their positions to other features, which allows for detection of higher order patterns (Lecun et al., 1998).

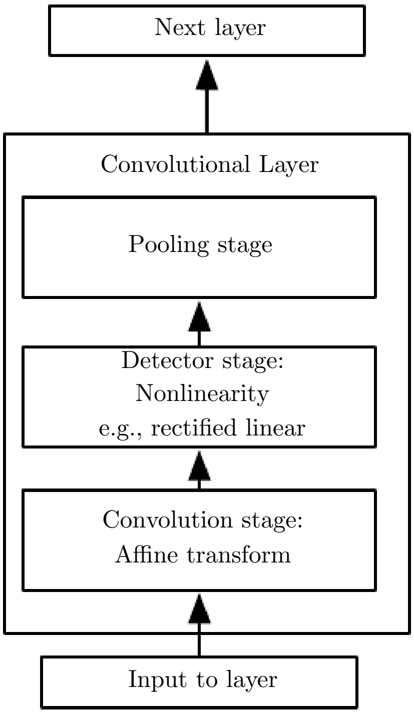


Figure 3.3. *The layout of a typical convolutional neural network layer. Image from Goodfellow et al. (2016, p. 341).*

Instead of scalar weights, a 2D-CNN contains 2D planes for weights. These weight planes, also known as filter kernels, map an input to a feature map using linear convolution. Nonlinearity is then applied after the linear convolution to make learning of nonlinear feature maps possible. Afterwards, feature maps are usually sub-sampled with a pooling stage to reduce the dimensions of the feature map (Kiranyaz et al., 2021). Pooling makes the feature representations less susceptible to small changes in the input. The last layer in a CNN is usually a fully-connected layer, which combines the feature maps for classification or regression. A typical convolutional layer is depicted in Figure 3.3 (Goodfellow et al., 2016, pp. 330–347).

Although CNNs are typically used for 2D data, they can be adapted for 1D time-series data as well (Goodfellow et al., 2016, pp. 330–339). The main difference between 1D and 2D CNNs is the dimensions of the kernel and feature maps. In 1D CNNs, the 2D matrices are replaced by 1D arrays (Kiranyaz et al., 2021).

Because the filter kernel is usually smaller than the input, CNNs have sparse connectivity instead of full connectivity like in MLPs. Sparse connectivity makes CNNs more efficient computationally and statistically, while using less memory to store parameters (Goodfellow et al., 2016, pp. 330–339).

4. EXPERIMENTS

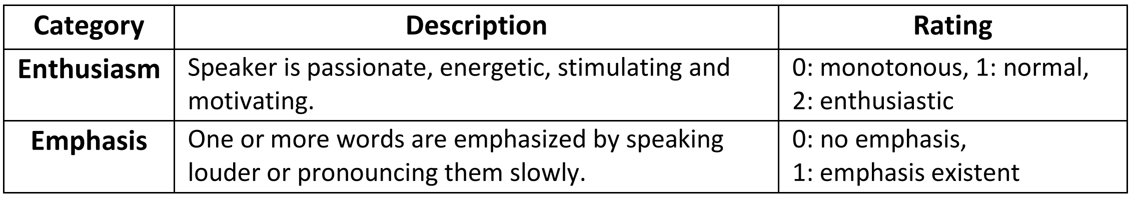
In this chapter, the present experiments are described in further detail. Section 4.1 depicts the dataset used in the present study, while Section 4.2 describes the experimental setup. The experiments were conducted using the features and classifiers described in Chapter 3.

4.1 Entheos dataset

The Entheos dataset (Viegas and Alikhani, 2021) consists of randomly selected TED talks from the TEDLIUM corpus release 3, which has audio of over 2000 talks. The talks were segmented into sentences by utilizing a transcript obtained with the Google cloud transcription service. The audio segments were matched with corresponding video segments obtained from the official TED website. All of the talks were in English. Noisy samples, like samples containing clapping or laughter, were discarded.

Multiple different labels were compared to define what labels to use for annotation. Ultimately, *enthusiasm* and *emphasis*, were selected based on their high inter-rater agreement scores.

Table 4.1. *Selected labels as defined in Entheos, reproduced from Viegas and Alikhani (2021).*



Category	Description	Rating
Enthusiasm	Speaker is passionate, energetic, stimulating and motivating.	0: monotonous, 1: normal, 2: enthusiastic
Emphasis	One or more words are emphasized by speaking louder or pronouncing them slowly.	0: no emphasis, 1: emphasis existent

The descriptions of enthusiasm and emphasis that were given to the annotators can be seen from Table 4.1. Additionally, Table 4.1 contains the possible levels for enthusiasm and emphasis. Enthusiasm was assigned to one of three levels, while emphasis was labeled as existent or not. Originally 1819 segments were labeled, of which 1126 were kept due to having more than one annotation. Altogether, these 1126 segments contained 60 male and 53 female speakers. The multiclass classification was converted into a separate binary classification by combining the "monotonous" and "normal" categories into a category called "non-enthusiastic". In the present study, only the labels regarding enthusiasm were used.

Training and test sets were created, with 1018 and 108 samples, respectively. The test set has talks from 5 male and 5 female speakers, while the training set has talks from 55 male and 48 female speakers. There is no speaker overlap between the training and test sets. A near equal distribution of genders is important due to the perception of enthusiasm being dependent on the gender of the speaker (Viegas and Alikhani, 2021).

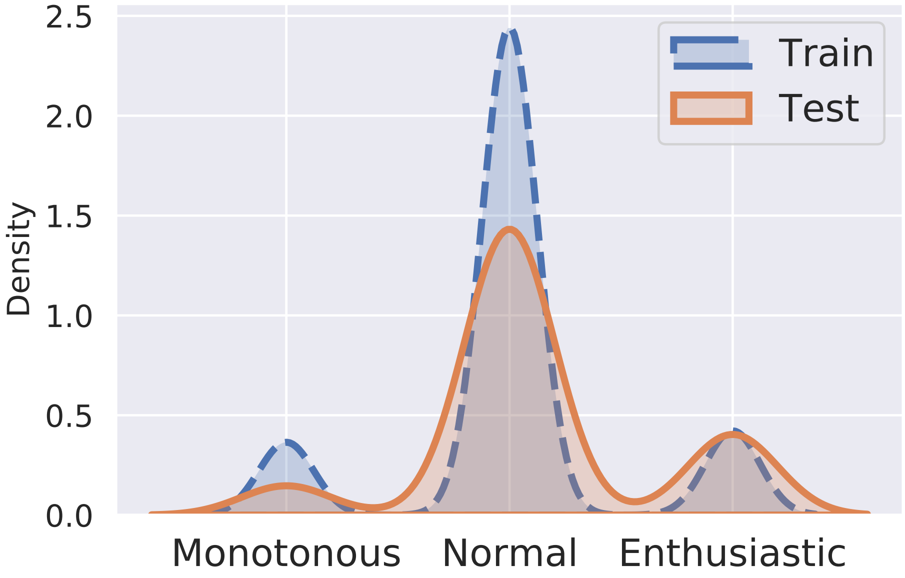


Figure 4.1. *The distribution of enthusiasm labels in training and test sets. Image from Viegas and Alikhani (2021).*

As can be seen from Figure 4.1, the class distribution of enthusiasm is uneven in both the training and the test set. Different types of features were extracted from the collected segments and compared with each other. The best-performing features were combined into a multimodal dataset.

4.2 Experimental setup

The existing eGeMAPS features, with corresponding two- and multiclass labels, were downloaded and imported into the experimental environment, along with the pre-determined train and test split (described in Section 4.1). Next, the features were preprocessed (Figure 3.1). The feature columns were standardized to a mean of zero and a standard deviation of one. Finally, class weights were computed as a relationship between the class occurrences and the total number of samples.

The models described in Section 3.2 were evaluated with precision, recall and F1-score. These metrics were chosen due to them being used in the Entheos paper (Viegas and Alikhani, 2021). As in Entheos, weighted averaging was used with these metrics to account for uneven distribution of the labels. Equations for binary classification and brief descriptions are shown for the metrics in Table 4.2 below.

Table 4.2. *Metrics which were used to evaluate the performance of the tested models in the present experiments. TP, FP and FN stand for true positives, false positives and false negatives, respectively. Equations from Schuller and Batliner (2013), descriptions from Goodfellow et al. (2016).*

Metric	Equation (binary)	Description
Precision (PR)	$\frac{TP}{TP+FP}$	Fraction of detections that were correct.
Recall (RE)	$\frac{TP}{TP+FN}$	Fraction of true events that were detected.
F1-score (F_1)	$2\frac{RE \cdot PR}{RE+PR}$	Harmonic mean of precision and recall.

All of the classifiers tested were compared to baseline results obtained by Viegas and Alikhani (2021) for eGeMAPS features. Baseline results were achieved with four fully-connected layers with ReLU activation functions, Adam optimizer and cross entropy loss as the loss function.

SVMs were implemented with scikit-learn, a machine learning library for Python. Preliminary tests showed that an RBF was the best-performing kernel function. The margin parameter C and the kernel scale parameter γ were determined iteratively. In each iteration the hyperparameter values were changed and the SVM was evaluated with 5-fold cross-validation. The best set of hyperparameters were chosen by calculating an average of the three evaluation metrics and saving the best result. The same average was used to optimize the SVM. An SVM was then trained with the best performing hyperparameteres, followed by evaluating the trained SVM on the test set.

Multiple MLP and CNN architectures were also tested. Performance is reported only for the best-performing MLP and CNN architectures in Chapter 5. The tested MLPs and CNNs were implemented using Python-based PyTorch. The predetermined training data was randomly split into a training set and a validation set, which were used in a training-validation loop. In the loop, the networks were trained using the training set and validation error was calculated with the validation set. Training was terminated based on a patience counter of 40 epochs if validation loss didn't improve. Afterwards, model state with the lowest validation loss was selected for evaluation with the test set. Adam optimizer was used during training and cross entropy loss was used to calculate the validation error.

The best-performing MLP consisted of an input layer, three hidden layers with 256 neurons each and an output layer. The output layer's size was 2 in binary classification and 3 in multiclass classification. The network had ReLU activation functions and dropout layer with 30% random dropout after every hidden layer. The CNN had two 1D convolutional layers. The first one had 64 kernels of size 7 and a stride of 2. The second convolutional layer had 32 kernels of size 5 and a stride 2. The output layer was a fully-connected layer with a size of 2 or 3, depending on the type of classification. The convolutional layers also had ReLU activation functions and 1D maxpooling layers with a kernel size of 2. A 20% random dropout was applied after the second 1D convolutional layer.

5. RESULTS

This chapter describes the last section of pipeline (Figure 3.1). The results of the present experiments are shown in Table 5.1, along with the baseline result of Viegas and Alikhani (2021), obtained with eGeMAPS features, which was the best result of acoustic-only features. In addition to the acoustic-only baseline, the best-performing multimodal features are displayed. The multimodal features consisted of one set of acoustic features and three different sets of text based features.

Table 5.1. *Precision, recall and F1-score for all of the tested models and the baseline model in binary (B) and multiclass (M) classification. The best scores for each metric in binary and multiclass classification are highlighted in grey.*

Classifier	Precision [B/M]	Recall [B/M]	F1-score [B/M]
SVM	0.89/0.73	0.86/0.74	0.87/0.73
MLP	**0.91**/**0.80**	**0.89**/0.76	**0.90**/0.77
CNN	0.89/**0.80**	0.87/**0.78**	0.88/**0.79**
Entheos (eGeMAPS)	0.80/0.59	0.71/0.47	0.74/0.50
Entheos (multimodal)	0.83/0.63	0.84/0.65	0.83/0.64

Based on the results, the best-performing classifier in binary classification was the MLP, while the CNN performed the best overall in multiclass classification. SVM was clearly the worst-performing out of the architectures in multiclass classification, but achieved scores close to the neural networks. Although SVM performed the worst out of the experimented classification models, it still achieved better scores than the baseline network in both binary and multiclass classification, regardless of the features used by the baseline network. The baseline network was clearly outperformed by the MLP and CNN, even when the baseline network was using features with different modalities, compared to the acoustic-only features used by the MLP and CNN. In our experiments, we observed that the standardization of input features contributed most to the increase in classifier performance compared to the baseline results of Viegas and Alikhani (2021), although model choice and architecture also had an impact. The results between the CNN and the MLP are extremely close, which puts more weight on the computational efficiency of CNNs.

6. CONCLUSION

Enthusiasm is an important part of teaching and engaging communication. Given the importance of enthusiasm, researchers are trying to develop enthusiastic robots and virtual agents. This thesis' aim was to experiment if different classifiers could outperform the baseline model given in the recently published Entheos dataset (Viegas and Alikhani, 2021). In the experiments, MLP, CNN and SVM were tested. Results showed that the MLP was the best at distinguishing enthusiastic speech from monotonous speech, while the CNN was the best at differentiating different levels of enthusiasm. SVMs had the worst performance but all classifiers achieved better results than the baseline network. SVMs are still a worthwhile option, especially in less comprehensive corpora. Using acoustic-only features, we outperformed the baseline network regardless of the features (multimodal or acoustic-only) used by the baseline network.

Although the dataset is practical regarding the gender diversity of the speakers and the quality of the recordings, the results might not be representative of real world performance. This is due to the dataset being collected from well-rehearsed speeches, which is a very specific context. The speeches all being in English could affect the results, due to disparities in the representation of emotional states between languages (Saad et al., 2021). Enthusiasm might manifest in different ways depending on the context and culture. Expanding the dataset to different contexts would improve the generalization of the examined models.

Also, the dataset used was relatively small and unevenly distributed regarding enthusiasm levels. A bigger and a more comprehensive dataset could lead to improvements in classifier performance. More complicated neural network architectures could also be examined with a larger dataset, as deeper architectures have achieved good performance in SER in the past (Fayek et al., 2017).

REFERENCES

Abe, S. (2010). *Support vector machines for pattern classification*. 2nd ed. Advances in Pattern Recognition. Springer.

Baevski, A., Hsu, W.-N., CONNEAU, A. and Auli, M. (2021). Unsupervised Speech Recognition. *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., pp. 27826–27839.

Boser, B. E., Guyon, I. M. and Vapnik, V. N. (1992). A Training Algorithm for Optimal Margin Classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT '92. Association for Computing Machinery, pp. 144–152.

Bostan, L.-A.-M. and Klinger, R. (2018). An Analysis of Annotated Corpora for Emotion Classification in Text. *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, pp. 2104–2119.

Daido, R., Ito, M., Makino, S. and Ito, A. (2014). Automatic evaluation of singing enthusiasm for karaoke. *Computer speech & language* 28.2, pp. 501–517.

Demszky, D., Movshovitz-Attias, D., Ko, J., Cowen, A., Nemade, G. and Ravi, S. (2020). GoEmotions: A Dataset of Fine-Grained Emotions.

Du, K.-L. (2014). *Neural Networks and Statistical Learning*. 1st ed. 2014. Springer London.

El Ayadi, M., Kamel, M. S. and Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern recognition* 44.3, pp. 572–587.

Eyben, F., Scherer, K. R., Schuller, B. W., Sundberg, J., Andre, E., Busso, C., Devillers, L. Y., Epps, J., Laukka, P., Narayanan, S. S. and Truong, K. P. (2016). The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing. *IEEE transactions on affective computing* 7.2, pp. 190–202.

Fayek, H. M., Lech, M. and Cavedon, L. (2017). Evaluating deep learning architectures for Speech Emotion Recognition. *Neural networks* 92, pp. 60–68.

Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep Learning*. `http://www.deeplearningbook.org`. MIT Press.

Haider, F., Fuente, S. de la and Luz, S. (2020). An Assessment of Paralinguistic Acoustic Features for Detection of Alzheimer's Dementia in Spontaneous Speech. *IEEE journal of selected topics in signal processing* 14.2, pp. 272–281.

Inaba, M., Toriumi, F. and Ishii, K. (2011). Automatic detection of "enthusiasm" in non-task-oriented dialogues using word co-occurrence. *2011 IEEE Workshop on Affective Computational Intelligence (WACI)*. IEEE, pp. 1–7.

Kasuriya, S., Theeramunkong, T., Wutiwiwatchai, C. and Sukhummek, P. (2018). Developing a Thai emotional speech corpus from Lakorn (EMOLA). *Language resources and evaluation* 53.1, pp. 17–55.

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M. and Inman, D. J. (2021). 1D Convolutional Neural Networks and Applications - A Survey.

Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Murphy, K. P. (2012). *Machine learning a probabilistic perspective*. Adaptive computation and machine learning series. MIT Press.

Nwankpa, C., Ijomah, W., Gachagan, A. and Marshall, S. (2018). Activation Functions: Comparison of trends in Practice and Research for Deep Learning. *CoRR* abs/1811.03378.

Rosenblatt, F. (1962). *PRINCIPLES OF NEURODYNAMICS. PERCEPTRONS AND THE THEORY OF BRAIN MECHANISMS*.

Saad, F., Mahmud, H., Shaheen, M. A., Hasan, M. K. and Farastu, P. (2021). Is Speech Emotion Recognition Language-Independent? Analysis of English and Bangla Languages using Language-Independent Vocal Features. *CoRR*.

Schuller, B. and Batliner, A. (2013). *Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing*. John Wiley & Sons, Incorporated.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. 15.1, pp. 1929–1958.

Swain, M., Routray, A. and Kabisatpathy, P. (2018). Databases, features and classifiers for speech emotion recognition: a review. *International journal of speech technology* 21.1, pp. 93–120.

Tang, Y. (2013). Deep Learning using Linear Support Vector Machines.

Tokuhisa, R. and Terashima, R. (2006). Relationship between Utterances and “Enthusiasm” in Non-task-oriented Conversational Dialogue. *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*. Association for Computational Linguistics, pp. 161–167.

Viegas, C. and Alikhani, M. (2021). Entheos: A Multimodal Dataset for Studying Enthusiasm. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, pp. 2047–2060.

Wani, T. M., Gunawan, T. S., Qadri, S. A. A., Kartiwi, M. and Ambikairajah, E. (2021). A Comprehensive Review of Speech Emotion Recognition Systems. *IEEE access* 9, pp. 47795–47814.