

**Speech Emotion Recognition**

CZ4042 Neural Networks & Deep Learning

Final Project

Submitted by:

Koh En Rong – U1822302K

Danson Kang Yit Siang – U1821068E

Cecelia Soh Yan Pei – U

School of Computer Science and Engineering

Contents

[Introduction 3](#_Toc86939809)

[Objective 3](#_Toc86939810)

[Methodology 3](#_Toc86939811)

[Database 3](#_Toc86939812)

[Features 3](#_Toc86939813)

[Classifier 3](#_Toc86939814)

[Experiments & Results 3](#_Toc86939815)

[Conclusion 3](#_Toc86939816)

# Introduction

The most basic ways of human expression are speech and body language. We depend on these recognitions to determine emotions, which plays a vital role in communications. As the technology advancement in the past few decades produced artificial intelligence that can perform speech recognition such as Siri and Alexa or facial recognition systems, human has been trying to extend the understanding of emotions to computers in various ways.

Since emotions are subjective and have different understandings depending on individuals, emotion detection has always been challenging. There is no standard agreement on measuring or categorising them but are evaluated by various factors such as knowledge or cultures, which others can easily misunderstand. As a result, the crucial task of automatic recognition of human emotions became a research field of significant interest.

Among all sources of emotion recognition such as facial expressions, physiological measurements, or communication content, speech signals are the most convenient and effective method since it is easy to acquire and non-invasive. It is also generally known that emotional changes are subconsciously reflected in speech utterances. Therefore more research can be done on this domain to better improve how AI understand emotions.

# Objective

In this project, we aim to determine if the tone of a person’s speech can be correctly mapped to the different emotions: Anger, Sadness, Happiness, Fear, Disgust, Surprised, and Neutral. We propose the use of a deep learning classifier compared to the common traditional classifier such as Hidden Markov Model or Support Vector Machine. This is done by extracting, experimenting and training with various time-series signal features and methods. These insights will then produce the most optimal results, which in turn be used to detect the speaker’s emotions in real-time.

# Literature Review

## Existing Techniques

Previous researches done on speech emotion recognition are discussed in this section. Naturally, most of the methods used are typically the traditional classifiers and did not include the recent trends and advances of deep neural networks such as Convolutional Neural Network or Recurrent Neural Network.

## Database

### Emo-DB

Emo-DB is a freely available German emotional database that is created by the Institute of Communication Science, Technical University, Berlin, Germany.

### RAVDESS Emotional speech audio

RAVDESS is also a freely available emotional database. It contains audio recordings in a North American accent.

# Methodology

## Features Extraction

## We used openSMILE – an open-source toolkit for audio feature extraction. This toolkit was a project started in 2008 at the Technical University Munich (TUM). For this project, we used the openSMILE Python library to extract the features needed to execute this project. The feature sets available in the Python library are as follows:

1. ComParE\_2016
2. GeMAPS *(deprecated)*
3. GeMAPSv01a
4. GeMAPSv01b
5. eGeMAPS *(deprecated)*
6. eGeMAPSv01a
7. eGeMAPSv01b *(deprecated)*
8. eGeMAPSv02
9. emobase

However, the feature sets GeMAPS, eGeMAPSv01b, and eGeMAPS are deprecated. Therefore for the future experiments, we will exclude the usage of these feature sets.

## Classifier

# Experiments & Results

* 1. Model

We experimented with various models to find out which model yielded better results. We are using the sequential API to build our models.

* + 1. Model

model = Sequential()

model.add(Conv1D(256, kernel\_size=8, strides = 1, activation='relu', input\_shape=(num\_of\_features,1)))

model.add(BatchNormalization())

model.add(MaxPooling1D(pool\_size=2))

model.add(Dropout(0.5))

model.add(Conv1D(128, kernel\_size=8, strides = 1, activation='relu'))

model.add(BatchNormalization())

model.add(MaxPooling1D(pool\_size=2))

model.add(Dropout(0.5))

model.add(Conv1D(64, kernel\_size=8, strides = 1, activation='relu'))

model.add(BatchNormalization())

model.add(MaxPooling1D(pool\_size=2))

model.add(Dropout(0.7))

model.add(Flatten())

model.add(Dense(64, activation='relu'))

model.add(Dense(32, activation='relu'))

model.add(Dense(9, activation='softmax'))

* 1. Effectiveness of feature sets

By comparing the val\_acc obtained by training the model using datasets with different feature combinations, the following results have been obtained.

A picture containing chart

Description automatically generated

It is observed that the emobase and ComParE\_2016 feature sets greatly outperformed the eGeMAPS datasets.

# Conclusion