Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Cybernetics

Multimodal Speech Emotion Recognition

Bachelor Thesis

Jan Čuhel

Study programme: Open Informatics
Specialization: Artificial Intelligence and Computer Science
Supervisor: Ing. Jan Pichl

Prague, May 2021

Project Supervisor:
Ing. Jan Pichl
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
Technická 2
160 00 Prague 6
Czech Republic
pichljan@fel.cvut.cz

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Čuhel Jan** Personal ID number: **483634**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Open Informatics**

Specialisation: **Artificial Intelligence and Computer Science**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Multimodal Speech Emotion Recognition

Bachelor's thesis title in Czech:

Multimodální rozpoznávání emocí z řeči

Guidelines:

The bachelor thesis will comprise the following steps:
- Research the available datasets for the task of emotion recognition
- Select a suitable set of emotions for classification based on available data
- Research existing solutions dealing with emotion recognition from text and audio
- Design architectures for emotion recognition systems for text inputs, audio inputs, and a combination of both
- Implementation and train proposed model architectures
- Test and evaluate the system results

Bibliography / sources:

[1] Gaurav Sahu - Multimodal Speech Emotion Recognition and Ambiguity Resolution - University of Waterloo , Ontario, Canada - 2019.
[2] Seunghyun Yoon, Seokhyun Byun, Kyomin Jung - Multimodal Speech Emotion Recognition Using Audio and Text - Seoul National University, Seoul, Korea - 2018.
[3] Kannan Venkataramanan, Haresh Rengaraj Rajamohan - Emotion Recognition from Speech - New York University, USA - 2018.
[4] Kashfia Sailunaz, Manmeet Dhaliwal, Jon Rokne & Reda Alhajj - Emotion detection from text and speech: a survey - University of Calgary, Calgary, Canada - 2018.
[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova - BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding - Google - 2019.

Name and workplace of bachelor's thesis supervisor:

Ing. Jan Pichl, Big Data and Cloud Computing, CIIRC

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **07.01.2021** Deadline for bachelor thesis submission: **21.05.2021**

Assignment valid until: **30.09.2022**

Ing. Jan Pichl	prof. Ing. Tomáš Svoboda, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
Supervisor's signature	Head of department's signature	Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

. _____
Date of assignment receipt

_____
Student's signature

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, date ....
Signature

Abstract

Abstract (en)

This work focuses on the Emotion Recognition task, which falls into the Natural Language Processing problems. The goal of this work was to create Machine learning models to recognize emotions from text and audio. The work introduces the problem, possible emotion representations, available datasets, and existing solutions to a reader. It then describes our proposed solutions for Text Emotion Recognition (TER), Speech Emotion Recognition (SER), and Multimodal Speech Emotion Recognition tasks. Further, we describe the experiments we have conducted, present the results of those experiments, and show our two demo practical applications. Two of our proposed models were able to outperform a previous state-of-the-art solution from 2018. All experiments and models were programmed in the Python programming language.

Keywords: Multimodal Speech Emotion Recognition, Emotion Recognition (ER), Text Emotion Recognition (TER), Speech Emotion Recognition (SER), Emotions, Natural Language Processing (NLP), Machine learning, Classification, Python.

Abstrakt (cs)

Tato práce se zaměřuje na problém Rozpoznávání emocí, který spadá do třídy problémů Zpracování přirozeného jazyka. Cílem této práce bylo vytvořit modely strojového učení na rozpoznání emocí z textu a ze zvuku. Práce základně seznámí čtenáře s tímto problémem, s možnostmi reprezentace emocí, s dostupnými datovými sadami a s existujícími řešeními. Poté se v práci popisují naše navrhnutá řešení pro úlohy Rozpoznávání emocí z textu, Rozpoznávání emocí ze zvuku a Multimodálního rozpoznávání emocí z řeči. Dále popisujeme experimenty, které jsme provedli, prezentujeme dosažené výsledky těchto experimentů a ukazujeme naše dvě praktické demo aplikace. Dva z našich navrhovaných modelů porazily předchozí nejlepší dostupné řešení z roku 2018. Všechny experimenty a modely byly naprogramovány v programovacím jazyce Python.

Klíčová slova: Multimodální rozpoznávání emocí z řeči, Rozpoznávání emocí, Rozpoznávání emocí z textu, Rozpoznávání emocí z řeči, Emoce, Zpracování přirozeného jazyka, Strojové učení, Klasifikace, Python.

Acknowledgements

First of all, I would like to express my enormous gratitude to my thesis supervisor, Ing. Jan Pichl. He has been a constant source of encouragement and insight during my research and helped me with numerous problems. Special thanks go to the staff of the Department of Cybernetics, CIIRC, and Team Alquist, who maintained a pleasant and flexible environment for my research.

List of Tables

List of Figures

List of Acronyms

ANNs Artificial Neural Networks. 15

ASR Automatic Speech Recognition. 14

BERT Bidirectional Encoder Representations from Transformers. 5

BoW Bag-of-Words. 5

CNNs Convolutional Neural Networks. 6

IEMOCAP Interactive emotional dyadic motion capture database. 16

LinSVM Linear Support Vector Machines. 7

LR Logistic Regression. 8

LSTM Long Short Term Memory. 5

MER Multimodal Emotion Recognition. 6

MFCCs Mel Frequency Cepstral Coefficents. 6

MLE Maximum Likelihood Estimation. 9

MNB Multinomial Naive Bayes. 8, 31

NLP Natural Language Processing. 1

RAVDESS Ryerson Audio-Visual Database of Emotional Speech and Song. 13

RF Random Forest. 8

RNNs Recurrent Neural Networks. 6

SER Speech Emotion Recognition. 5

SOTA state-of-the-art. 26, 29

TER Text Emotion Recognition. 5

TFIDF Term Frequency-Inverse Document Frequency. 5

TRILL TRIpLet Loss network. 14

Contents

Chapter 1

Introduction

1.1 About

In this thesis, we focus on the Emotion Recognition task. Emotion Recognition is a fascinating area of Natural Language Processing (NLP), and it is still an active area of research. NLP is a sub-field of Machine learning that deals with the interaction between computers and natural language data. Its objective is to try to understand the data. Emotions and dealing with them are a massive part of human life. As humans, we are not entirely rational (in contrast to computers). We act based on our emotional state, so computers must understand our emotions to make the interaction between humans and machines more comfortable and intuitive. Emotion Recognition models can help in this matter. They try to recognize emotion(s) based on an input to help a machine act based on it. Emotion Recognition has many use-cases from which some of them are already applied in real life. For example, the use-cases include security cameras in shops capturing customers' faces to recognize their emotions and, based on that, set the right genre of music, or social bots recognizing the people's emotion to display a kind of empathy, and many more.

1.2 Goal of the thesis

The goal of this thesis is to develop a Machine learning model for Emotion Recognition from text and audio (speech). The thesis has several steps, which are following:

- Research and acquaintance with datasets for the task of Emotion Recognition
- Selection of a suitable set of emotions for classification based on available data
- Research of existing solutions dealing with Emotion Recognition from text and audio
- Design of architectures for Emotion Recognition systems for text inputs, audio inputs, and a combination of both
- System implementation and training

- Testing and measuring the system results

We have followed these steps during working on the thesis.

1.3 Structure of the report

This report is organized into 6 chapters as follows:

1. *Introduction*: Describes the motivation behind our efforts altogether with our goals.
2. *Problem description*: Introduces to the reader the necessary theoretical background and some of the related work.
3. *Our Approach*: Describes our solution for the task.
4. *Experiments and Results*: Describes the experiments that we have conducted and shows their results.
5. *Practical applications*: Describes the practical applications we have created.
6. *Conclusion*: Summarizes the results of the thesis, suggests possible topics for further research, and concludes the work.

Chapter 2

Problem description

Emotion Recognition is a Machine learning task, which can be both regression or a classification problem. It depends on the emotional representation the model uses. Nevertheless, we tend to think that more common is the classification task. In the classification task, the model assigns each input an emotion or multiple emotions from a discrete set of emotions. It is very similar to the Sentiment Analysis problem, where the objective is to find a sentiment (positive, neutral, or negative) that best describes the input data. Sentiment Analysis is also known as Opinion Mining. Emotion Recognition is still a matter of progress. Therefore, there is room for improvement.

2.1 Emotion models

The question of representing emotions is an interesting topic that has many possible answers. We will briefly cover some of the ideas. All emotion models can be divided into two groups: Categorical and Dimensional [1]. Categorical emotion models define a discrete list of emotions, whereas Dimensional emotion models presuppose that emotional states are not independent and that they are related to each other. Two or three dimensions are usually used for describing an emotional state (**valence**, **arousal**, and **power**) [2, 3, 4].

Ekman's model Paul Ekman's basic emotions model [5] is probably the most famous Categorical emotion model there is. Paul Ekman recognized six fundamental emotions. These fundamental emotions are **happiness**, **sadness**, **anger**, **fear**, **surprise**, and **disgust**.

Dimensional emotion models We will mention two Dimensional emotion models; Plutchik's model [6] (shown in Figure 2.1) and Russell's model [7] (shown Figure 2.2). Both define a wheel shaped model.

[1] `https://upload.wikimedia.org/wikipedia/commons/archive/c/ce/20200104064132%21Plutchik-wheel.svg`

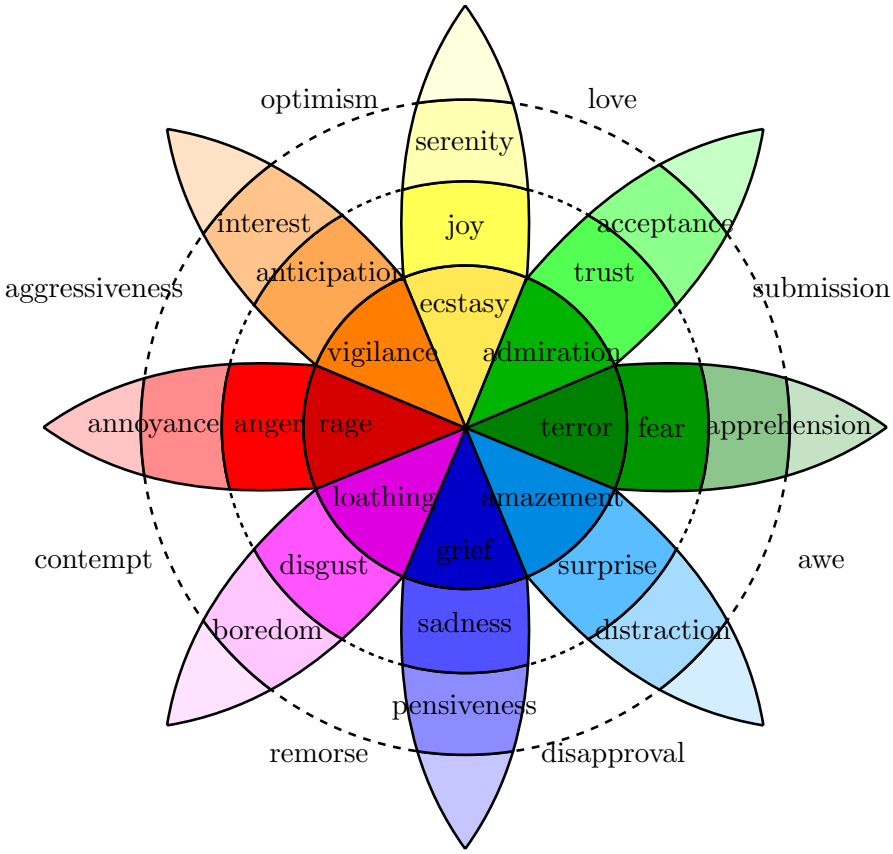

Figure 2.1: Plutchik's wheel of emotions[1]

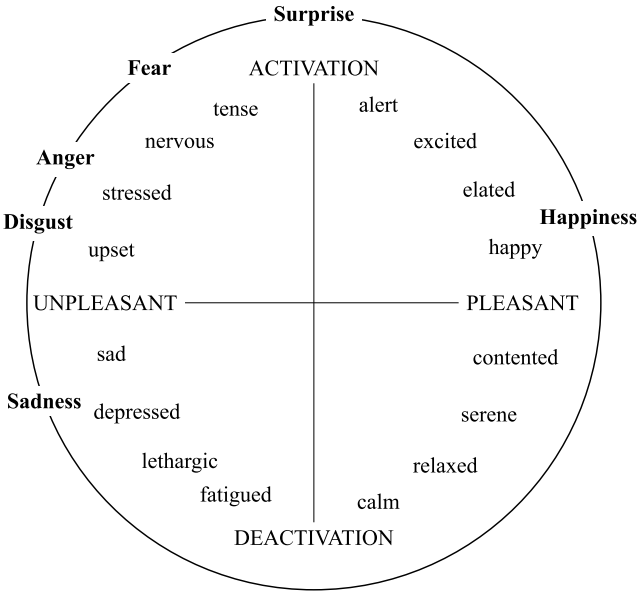

Figure 2.2: Russell's circumplex model of affects [8]

2.2 Types of Emotion Recognition

The type of Emotion Recognition depends on the type of input data. For text inputs, it is called Text Emotion Recognition. For audio (speech) inputs, it is called Speech Emotion Recognition,

and so on.

2.2.1 Text Emotion Recognition (TER)

As the name suggests, the Text Emotion Recognition (TER) is a task to recognize emotions from a text.

Available datasets

There are many available datasets. The most common datasets we have found are ISEAR [9], PsychExp [10], SemEval [11], and EmotionLines [12]. However, there are other available datasets like GoEmotions [13], DailyDialog [14], Alm [15], Aman [16], EmoBank [17], and CrowdFlower [18].

Key features

They are many types of features that can be used. Probably the most common are Bag-of-Words (BoW) [19], Term Frequency-Inverse Document Frequency (TFIDF) [20] and various kinds of Text Embeddings. The Text Embeddings break up into several groups based on what they represent. Essentially, they do the same thing; they take some text and represent it as a vector so that two vectors representing two texts with similar meaning should be closer to each other than two vectors representing two texts with a different meaning. Text Embeddings can be used either for words/tokens or whole sentences. The most common Word Embeddings are word2vec [21], GloVe [22], fasttext [23], GPT-2 [24], and Bidirectional Encoder Representations from Transformers (BERT) [25].

Related work

As Deep learning's popularity has grown over the last few years, it became more common to use it on TER. Widely used are models based on Long Short Term Memory (LSTM)[26], one of them is the DeepMoji model [27]. The introduction of this model has taken a big step forward for the TER. Another huge step forward was the introduction of BERT [25] text representation, which is along with its various versions nowadays used in many state-of-the-art solutions across a large variety of tasks.

2.2.2 Speech Emotion Recognition (SER)

Speech Emotion Recognition (SER) is a task in which the models try to recognize emotions from audio recordings of speech.

Available datasets

There is a relatively large variety of available datasets for SER task. The most used is probably the RAVDESS dataset [28]. Other available datasets are TESS [29], SAVEE [30], CREMA-D

[31], Emo-DB [32], or SEMAINE [33].

Key features

Audio can be represented with many audio features. They can be time-based or frequency-based. They can trace, for example, a pitch, energy, or entropy. These features are worth mentioning Spectral Centroid [34], Tonnetz [35], Chroma [36], Mel Spectrograms [37], and Mel Frequency Cepstral Coefficents (MFCCs) [38]. There also exist Audio Embeddings; we came across wav2vec [39] developed by Facebook AI Research Team and TRILL[40]. Refer to 3.3.2 for more detailed descriptions of the audio features we have used in this thesis.

Related work

The traditional Machine learning models such as Support Vector Machines, which is used in this work [41], are being replaced by Deep learning models. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are widely used in many recent works [37, 42].

2.2.3 Multimodal Emotion Recognition (MER)

Multimodal Emotion Recognition (MER) is a Machine learning task, where the models recognize emotions from a combination of modalities, such as video, audio, text, etc. The most accurate models are the ones that use the most modalities, such as audio, text, and video inputs. However, they are also the most sophisticated and most complicated to implement. It is also not a trivial task to obtain all three inputs, so obviously, there are not many available datasets.

Available datasets

Most common datasets that we have discovered are IEMOCAP [43], CSU-MOSEI [44] and MELD [45]. There exist other multimodal datasets, but they are for the Sentiment Analysis task.

Key features

There are many possibilities of what can be used as features for MER models. The first possibility is to use features from each modality and simply concatenate them. However, there are also solutions where features of some modalities are combined somehow into a new feature.

Related work

Deep learning found its way to MER models as well. Its models are the most common MER solution at the moment [46, 47, 48] However, in Gaurav Sahu's work [49], traditional Machine learning models (Random Forest, Logistic Regression, Multinomial Naive Bayes, Support Vector Machines, etc.) and their combinations are used.

Chapter 3

Our Approach

We use several datasets, one multimodal and the others for each modality in our approach. All datasets are in English. Therefore, we developed several models for Emotion Recognition for the English language. We train and finetune our models on the multimodal dataset and then use the other datasets to measure the models' accuracy for comparison purposes. We started with baseline models and then continue with complex models for TER, SER and MER tasks.

3.1 Baseline Models

Throughout each modality, we utilized four different baseline Machine learning models altogether. We exploited Multinomial Naive Bayes, Linear Support Vector Machines, Logistic Regression, and Random Forest models. In the following four subsections, we cover the basics of each baseline model. For each task and each baseline model, we found the best combination of hyperparameters via 5-fold CrossValidation.

3.1.1 Linear Support Vector Machines

Linear Support Vector Machines (LinSVM) [50] is a type of Support Vector Machines that uses a linear kernel. It is a supervised model that solves an optimization problem to find a separating hyperplane that maximizes the margin between two classes. For multiclass classification problems, the one-vs-rest strategy can be used. It means that it will solve k optimization tasks, where k represents the number of classes (note $k > 2$). For the ith classifier, let one class be all the points in class i, and let the second class be all the remaining points. The predicted class is the class that maximizes the margin for all the classes (it has the biggest objective function value). The implementation inside of the Python library scikit-learn can be found here.[1]

We were searching for the best value of these hyperparameters:

- **multi_class** - defines which multiclass strategy to use, can be either one-vs-rest or Crammer-Singer, the default value is one-vs-rest
- **max_iter** - defines the maximum number of iterations to be run, the default value is 1000 iterations

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html`

- **loss** - specifies the loss function, options are hinge or squared hinge function, the default one is squared hinge function
- **class_weight** - set the weights for each class, options are either a dictionary or 'balanced' option, default value is 'None'
- **C** - defines the regularization parameter, the strength of the regularization is inversely proportional to C, its default value is 1.0

3.1.2 Random Forest

Random Forest (RF) [51, 52] is an ensemble model that fits multiple Decision Trees [53] at the training phase. For the regression task, it outputs the mean of all the trained Decision Trees outputs. The classification task uses a majority vote principle, so it outputs the class with the most votes from the trained Decision Trees. The implementation inside of the Python library scikit-learn can be found here.[2]

We were searching for the best value of these hyperparameters:

- **n_estimators** - defines the number of trees in the forest, the default number is 100
- **min_samples_split** - specifies the minimum number of samples required to split an internal node, the default value is 2
- **max_features** - specifies the number of features to consider when looking for the best split, options are an integer or float, or 'auto', or 'sqrt', or 'log2', the default option is the 'auto'
- **max_depth** - defines the maximum depth of a Decision Tree, by default, it is unlimited
- **class_weight** - set the weights for each class, options are either a dictionary, or a list of dictionaries, or 'balanced', or 'balanced_subsample' option, the default value is 'None'

3.1.3 Multinomial Naive Bayes

Multinomial Naive Bayes (MNB) [54] is one of the Naive Bayes classifiers. They are a family of generative classifiers based on applying Bayes' theorem and assuming that the features are independent. Under multinomial settings, the feature components x_i represent the frequencies with which some events have been observed/generated by a multinomial probability distribution $(p_1, \ldots, p_n)$, where p_i represents the probability with which an event i occurs. The implementation inside of the Python library scikit-learn can be found here.[3]

We searched for the best value of hyperparameter **alpha**, representing the additive Laplace smoothing parameter (0 for no smoothing). The default value is 1.0.

3.1.4 Logistic Regression

Logistic Regression (LR) [55] is a statistical model usually used for a binary classification. In a nutshell, it estimates the posterior conditional probability $p(y|\mathbf{x})$ by using the logistic sigmoid function

$$h(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}}.$$

[2] `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`

[3] `https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html`

It is a parametric and discriminative model. The Maximum Likelihood Estimation (MLE) principle is used to calculate the optimal weights **w**. For a multiclass classification task, either one-vs-rest strategy, or softmax function instead of logistic sigmoid function can be utilized. The implementation inside of the Python library scikit-learn can be found here.[4]

We were searching for the best value of these hyperparameters:

- **solver** - determinates which algorithm to use in the optimization problem, options are 'newton-cg', 'lbfgs', 'liblinear', 'sag', and 'saga', the default one is the 'lbfgs'
- **penalty** - specifies the norm which is used in the penalization, options are 'l1', 'l2', 'elasticnet' and 'none', the default penalty is the 'l2'
- **multi_class** - defines which multiclass strategy to use, can be either 'auto', or 'one-vs-all', or 'multinomial', the default option is the 'auto' option
- **max_iter** - the same as for the LinSVM, the default value is 100
- **class_weight** - the same as for the LinSVM
- **C** - the same as for the LinSVM

3.2 Text Emotion Recognition (TER)

3.2.1 Used Datasets

In this work, we have used the PsychExp dataset [10], which contains 7480 samples of text messages labeled into a total of 7 emotions (**joy**, **fear**, **anger**, **sadness**, **disgust**, **shame**, and **guilt**). The Figure 3.1 shows the emotion distribution of this dataset.

3.2.2 Used Features

In this work, we have chosen to use the TFIDF text feature and BERT and Electra small [56] Embeddings as text features. We exploited the TFIDF text feature in baseline models and BERT and Electra small Embeddings in our complex TER model.

TFIDF Term Frequency-Inverse Document Frequency (TFIDF) [20] is a statistical measurement used to evaluate how important a word is to a document in a corpus. It scales down the weights of tokens that frequently occur in a given corpus and, therefore, do not contain much new information. TFIDF consists of 2 parts: Term Frequency and Inverse Document Frequency. Term Frequency $tf(t, d)$ of a term t and a document d is computed as a number of the term's occurrences in the document divided by the document's total number of terms. Inverse Document Frequency $idf(t)$ of the term t is computed as follows:

$$idf(t) = \log(\frac{N}{k}),$$

where N is the total number of documents in the corpus and k is the number of documents containing term t. The final value $tfidf(t, d)$ for the term t and the document d is computed as

[4] `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html`

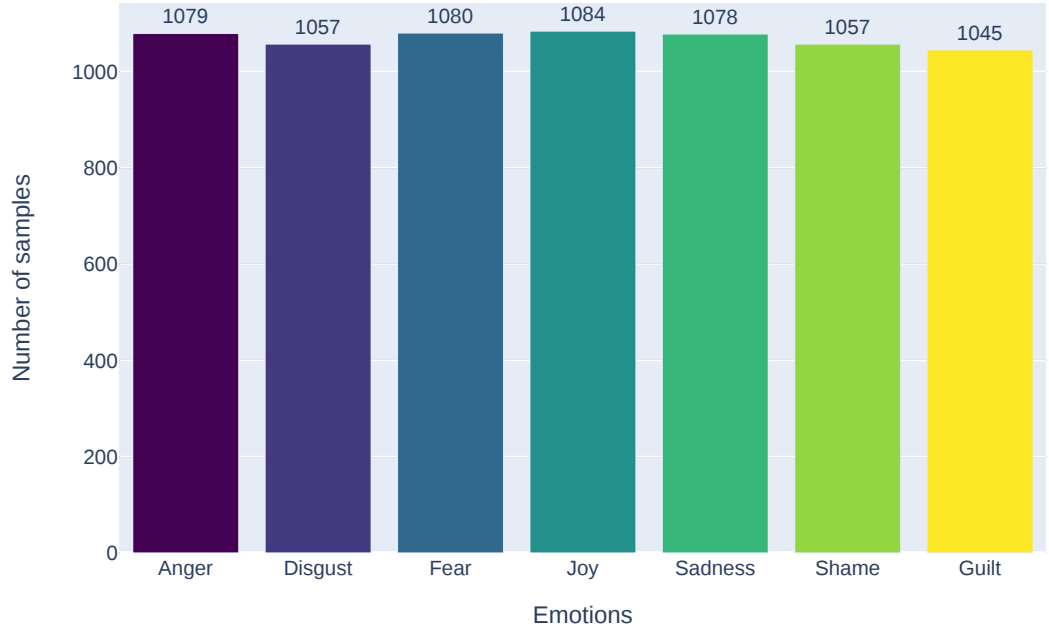

Figure 3.1: A figure showing number of samples for each emotion class in the PsychExp dataset

a product of $tf(t,d)$ and $idf(t)$[5]. The implementation inside of the Python library scikit-learn can be found here[6]. We finetuned some of the TFIDF hyperparameters. However, the most important hyperparameter turned out to be the **ngram_range**, which defines the range of n-grams that can be used. We found out that the best value is $(1,2)$. That means that unigrams and bigrams are used.

BERT Bidirectional Encoder Representations from Transformers (BERT) [25] is one of the Transformers models used to convert text inputs into a vector space. It is exploited in many state-of-the-art tasks across NLP problems and beyond. Its introduction was very revolutionary for the NLP field. In a nutshell, BERT takes preprocessed texts in the form of tokens and encodes them to a vector representation. Basically, it is a trained stack of Transformer Encoders [57]. There are many variants and implementations of different types of BERT models. We have selected the uncased version with 12 Transformers blocks. It uses a hidden size of 768 dimensions and 12 attention heads. This model has been pre-trained for English on Wikipedia and BooksCorpus. We then finetuned it for the TER task. The model was loaded from the TensorFlow Hub[7].

[5] `http://www.tfidf.com/`

[6] `https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html`

[7] `https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/4`

Electra Electra [56] is another version of Transformer based model used for language representation learning. Instead of replacing some tokens with a mask and train a model to predict the original tokens as BERT models do, Electra models use a more sample-efficient technique called replaced token detection. It means that Electra models are trained to recognize whether each input token was replaced by some generated one or not[8]. That makes the Electra models discriminators rather than generators as BERT models. Electra has three versions, small, base, and large. The pre-trained model was loaded from the TensorFlow Hub[9] and then finetuned for the TER task.

3.2.3 TER Models

Baseline Models

Implementation of the baseline TER models was inspired by these repositories[10][11]. Here are the base values of hyperparameters found for the baseline models for the TER task.

Linear SVM:

- max_iter: 800
- loss: 'squared_hinge'
- class_weight: None
- C: 0.068

Random Forest:

- n_estimators: 600
- min_samples_split: 26
- max_features: 'log2'
- max_depth: 24
- class_weight: 'balanced'

Multinomial NB:

- alpha: 0.56

Logistic Regression:

- solver: 'saga'
- penalty: 'l2'
- multi_class: 'auto'
- max_iter: 800
- class_weight: None
- C: 0.42

Complex Models

In our work, we have chosen to use two complex TER models. The first one uses BERT Embedding, followed by Bidirectional LSTM and Dense layers, and further, we will refer to this model simply as BERT TER model. The second one uses Electra small Embedding followed by Dense Layers, and further, we will refer to this model simply just as Electra small TER model. They were inspired by Classify text with BERT tutorial [58] and Working with Hugging Face Transformers and TF 2.0 article [59]. The Figure 3.2 shows the exact structure of the BERT TER model, where *seq_length* represents the number of the tokens that were encoded and returned. If the sentence has fewer tokens than the *seq_length*, it is padded. If the sentence is too long, it is truncated. The Figure 3.3 shows the structure of the Electra TER model, where the *CLS* token encodes the whole text sequence into a single vector. It is widely used for classification.

LSTM LSTM stands for Long Short Term Memory [26] and it is a type of Recurrent Neural Network (RNN). All RNN models follow the same structure. It is a sequence of RNN cells, which each outputs the result and sends the result as an input to the next RNN cell. RNNs

[8] `https://github.com/google-research/electra`
[9] `https://tfhub.dev/google/electra_small/2`
[10] `https://github.com/TetsumichiUmada/text2emoji`
[11] `https://github.com/Demfier/multimodal-speech-emotion-recognition`

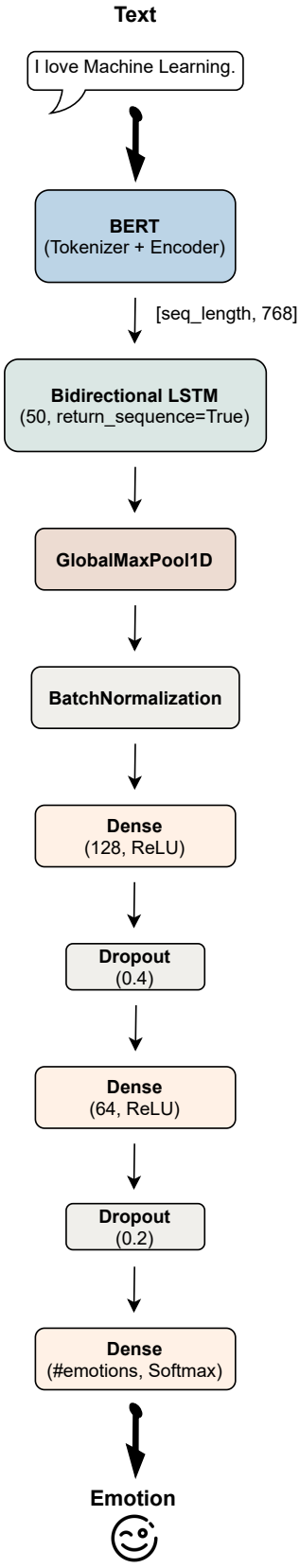

Figure 3.2: A structure of the complex TER model using BERT

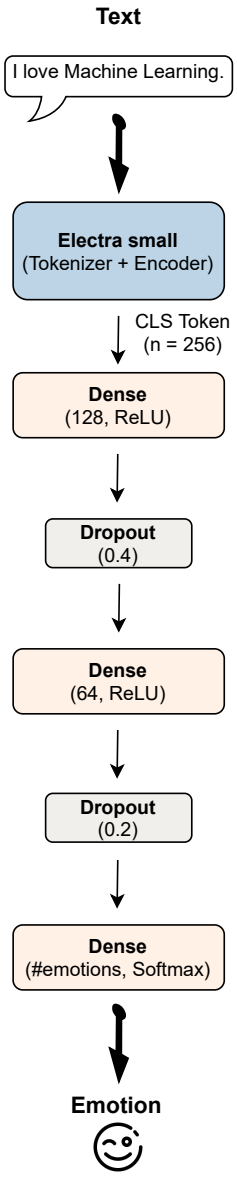

Figure 3.3: A structure of the complex TER model using Electra small

return a sequence of values, which often is not required, so only the value from the last cell is taken. LSTMs are capable of learning long-term dependencies, and for that, they are popular. LSTM cell contains 3 gates; an input gate, an output gate, and a forget gate.

Dense layer A Dense layer is when neurons in a layer are fully connected to the neurons from a previous layer.

Dropout Dropout [60] is a technique for addressing the problem of overfitting by randomly dropping some units (along with their connections) from the Neural Network during the training phase. This prevents units from co-adapting too much.

3.3 Speech Emotion Recognition (SER)

3.3.1 Used Datasets

In this work, we have used the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [28] dataset, which contains a total of 1440 audio recordings of speech from 24 professional actors (12 women and 12 men) and maps a total of 8 emotions (**neutral**, **calm**, **happy**, **sad**, **angry**, **fearful**, **disgusted** and **surprised**). The Figure 3.4 shows the emotion distribution of this dataset.

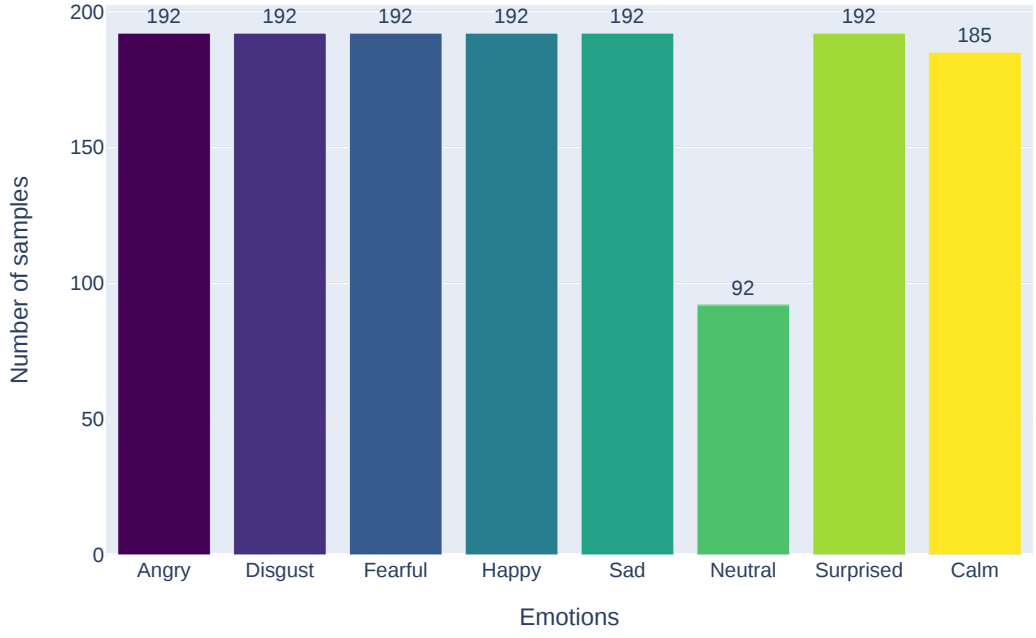

Figure 3.4: A figure showing number of samples for each emotion class in the RAVDESS dataset

3.3.2 Used Features

For this thesis, three audio features in total were chosen and two audio Embeddings. The audio features are the following MFCC, Mel Spectrogram, and Chroma. We then created every possible combination of these features (7 combinations in total). We ran an experiment to see which feature combination returns the best results across some of our SER models (more about the experiment and its results in 4.3.2). This best combination was then used in all those models. The audio Embeddings we chose to utilized are TRIpLet Loss network (TRILL) [40] and YAMNet [61]. In the following paragraphs, we explain these audio features and the audio Embeddings. The implementation of the audio features extraction was inspired by the Speech Emotion Recognition repository[12].

Mel Frequency Cepstral Coefficients (MFCC)

MFCCs introduced by Davis and Mermelstein [38] are widely used in Automatic Speech Recognition (ASR). The shape of the vocal tract (including the tongue, teeth, etc.) determines what sound is generated. This shape manifests itself in the envelope of the short-time power spectrum. MFFCs try to represent this envelope [62].

Mel Spectrogram

A Mel Spectrogram [37] is a spectrogram that has a Mel Scale on the y-axis. A spectogram plots a time vs amplitude on frequency graph of an audio signal [63].

Chroma

The Chroma feature [36] is a 12-dimensional audio feature vector. Each element represents a pitch class, and its value indicates how much energy of that pitch class is present in the signal on a standard chromatic scale [64].

TRILL

TRILL [40] is a pre-trained audio representation model that was trained on the AudioSet dataset [65]. Its objective is to represent audio parts closer together in time to be closer in the Embedding space by using the Triplet Loss function and other techniques from [66]. It uses the architecture from [67] which is inspired by the Image Classification model ResNet-50 [68], which means that TRILL is based on Convolutional Neural Networks.

YAMNet

YAMNet [61] is a pre-trained Deep learning audio classifier model that was trained on the AudioSet dataset [65]. The architecture of this model was inspired by the Image Classification

[12] https://github.com/x4nth055/pythoncode-tutorials/tree/master/machine-learning/speech-emotion-recognition

model MobileNet v1 [69], which also makes YAMNet CNN-based model. However, the model can also be used as an Audio Embedding, which is how we have used it because it takes an audio waveform as an input and frames it into small frames, so-called sliding windows. It then processes batches of these frames and returns Embeddings for each one of them. The model is loaded from TensorFlow Hub[13].

3.3.3 SER Models

Baseline Models

Implementation of the baseline SER models was inspired by How to Make a Speech Emotion Recognizer Using Python And Scikit-learn article [70] and the implementation of Multimodal Speech Emotion Recognition and Ambiguity Resolution [49]. Here are the base values of hyper-parameters found for the baseline models for the SER task.

Linear SVM:

- multi_class: 'ovr'
- max_iter: 1800
- loss: 'squared_hinge'
- class_weight: None
- C: 0.75

Random Forest:

- n_estimators: 600
- min_samples_split: 26
- max_features: 'log2'
- max_depth: 8
- class_weight: 'balanced_subsample'

Multinomial NB:

- alpha: 0.012

Logistic Regression:

- solver: 'newton-cg'
- penalty: 'l2'
- multi_class: 'auto'
- max_iter: 1500
- class_weight: None
- C: 0.58

Complex Models

In our work, we have chosen to use several complex SER models. The first one uses 1-dimensional Convolutional Neural Network, and further, we will refer to this model simply as 1D CNN SER model. The second one uses a 2-dimensional Convolutional Neural Network, and further, we will refer to this model simply as 2D CNN SER model. The third one uses TRILL Embedding, and further, we will refer to this model simply as TRILL SER model. The last one uses YAMNet as its high-level feature extractor, and further, we will refer to this model simply as YAMNet SER model. Each one of these models is shortly presented in the following paragraphs.

CNN Convolutional Neural Networks (CNNs, ConvNets) [71] are a subset of Artificial Neural Networks (ANNs). They are inspired by the Visual Cortex in the human brain. Individual neurons are dependent only on the neighboring neurons. That is why CNNs require fewer parameters than regular ANNs, making them easier to train. CNNs usually consist of Convolution, Pooling, and Fully Connected layers. They are usually used for huge inputs such as images. They are divided into different types depending on the dimensionality of their inputs, so for 1-dimensional inputs, the CNNs are called 1-dimensional CNNs (1D CNN in short) and so forth. The most common types are 1D CNNs, and 2D CNNs, mostly used for images.

[13] `https://tfhub.dev/google/yamnet/1`

1D CNN We used 2-layer 1-dimensional CNN followed by Dense layers model, which was inspired by Speech Emotion Analyzer repository[14]. The Figure 3.5 shows the exact structure of this model.

2D CNN We used 4-layer 2D CNN architecture followed by Dense layers. The model is divided into two 2D CNN blocks (as shown in Figure 3.7, the exact shapes are for the IEMOCAP dataset, the same model but trained on the RAVDESS dataset has a slightly different shapes), where each block consists of 2-layer 2D CNN (as shown in Figure 3.8). The model takes as an input 2D Log Mel Spectrograms. This architecture was inspired by Kannan Venkataramanan, and Haresh Rengaraj Rajamohan [72].

TRILL This model uses TRILL [40] to translate audio files into a vector representation. Then the vectors are passed into 4 LSTM layers followed by Dense layers. The Figure 3.6 shows the exact structure of this model, where *time* has a value of 59 for the audio files from the IEMOCAP dataset and value of 12 for the audio files from the RAVDESS dataset.

YAMNet This model has the same structure as the TRILL SER model (which is shown in the Figure 3.6, where n represents the number of frames for one audio file), but instead of TRILL it uses YAMNet to extract vector representation from input audio files.

3.4 Multimodal Emotion Recognition (MER)

3.4.1 Used Datasets

In this work, we have used the Interactive emotional dyadic motion capture database (IEMOCAP) [43], which contains approximately 12 hours of audiovisual data of improvised and scripted sessions from 10 actors (5 women and 5 men). Sessions are manually segmented into utterances. Human annotators then annotated each utterance. The dataset maps 11 emotional states (**angry**, **happy**, **sad**, **neutral**, **frustrated**, **excited**, **fearful**, **surprised**, **disgusted**, **other**, and **xxx** - undecided). The Figure 3.9 shows the emotion distribution of this dataset.

3.4.2 Used Features

We made use of the same features we had used in each modality before. We described each one of them in the previous sections.

[14] `https://github.com/MITESHPUTHRANNEU/Speech-Emotion-Analyzer`

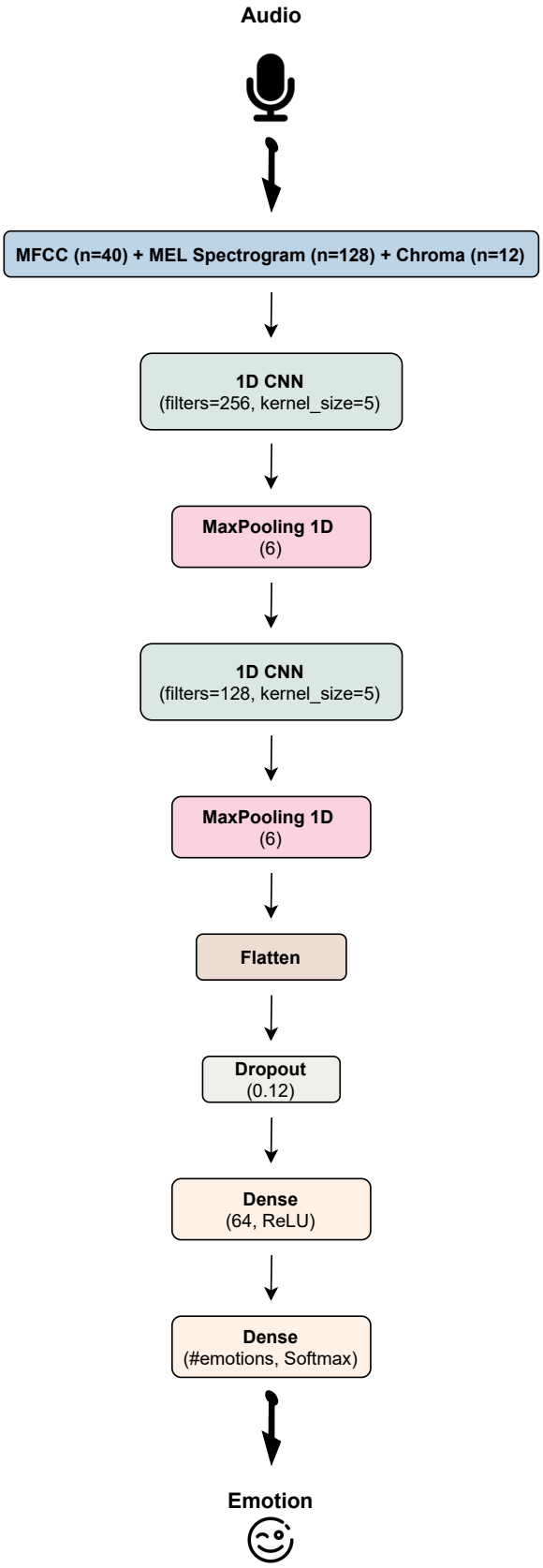

Figure 3.5: A structure of the complex SER model using 1D CNN

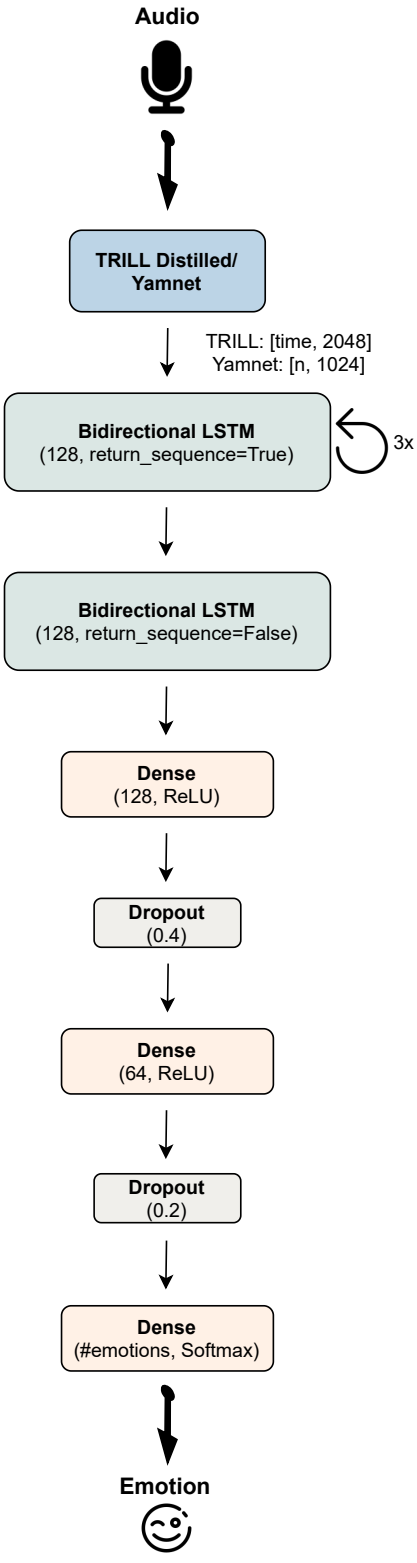

Figure 3.6: A structure of the complex SER model using TRILL or YAMNet, LSTM and Dense layers

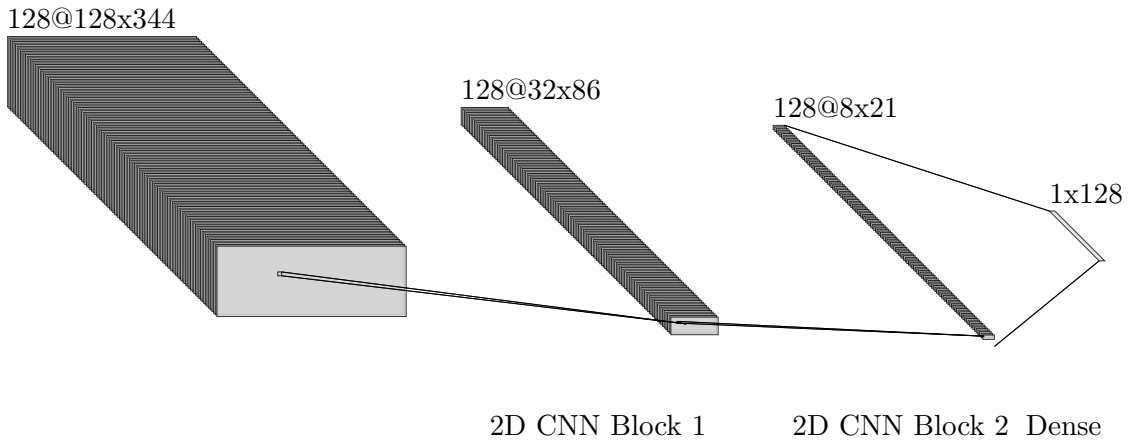

Figure 3.7: A structure of the complex SER model using 2D CNN

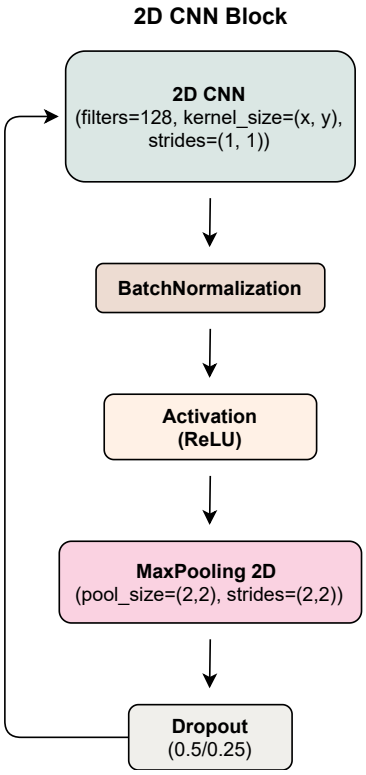

Figure 3.8: A structure of the 2D CNN Block

3.4.3 MER Models

Baseline Models

The implementation of the MER baseline models was inspired by this paper [49] (its implementation is available here[15]). Here are the base values of hyperparameters found for the baseline models for the MER task.

[15] `https://github.com/Demfier/multimodal-speech-emotion-recognition`

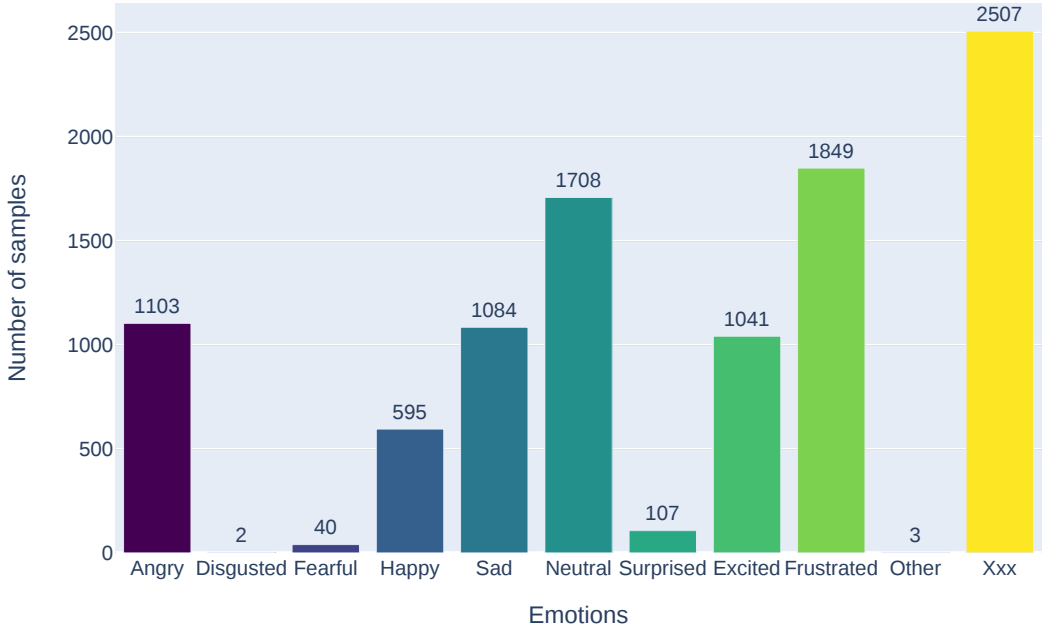

Figure 3.9: A figure showing number of samples for each emotion class in the IEMOCAP dataset

Linear SVM:

- multi_class: 'ovr'
- max_iter: 800
- loss: 'squared_hinge'
- class_weight: None
- C: 0.058

Random Forest:

- n_estimators: 600
- min_samples_split: 30
- max_features: 'log2'
- max_depth: 16
- class_weight: 'balanced'

Multinomial NB:

- alpha: 0.031

Logistic Regression:

- solver: 'newton-cg'
- penalty: 'l2'
- multi_class: 'auto'
- max_iter: 800
- class_weight: None
- C: 0.56

Complex Models

We took all the complex models from each modality (TER, SER) and joined them to create MER models. We stripped away the classification layer (Dense layers) from all complex TER and SER models and concatenated the text and audio features into a single vector which we passed into a classification layer. This process of concatenating before classification is called Early Fusion [73]. The classification layer is the same for all the MER models, except the MER model with Electra small and 1D CNN. The Figure 3.10 shows the architecture of Electra small with TRILL and Electra small with YAMNet MER models. We created only a single MER model with the BERT Embedding, and that is the one with 1D CNN for the audio part. In contrast, we used Electra small in combination with all the SER models.

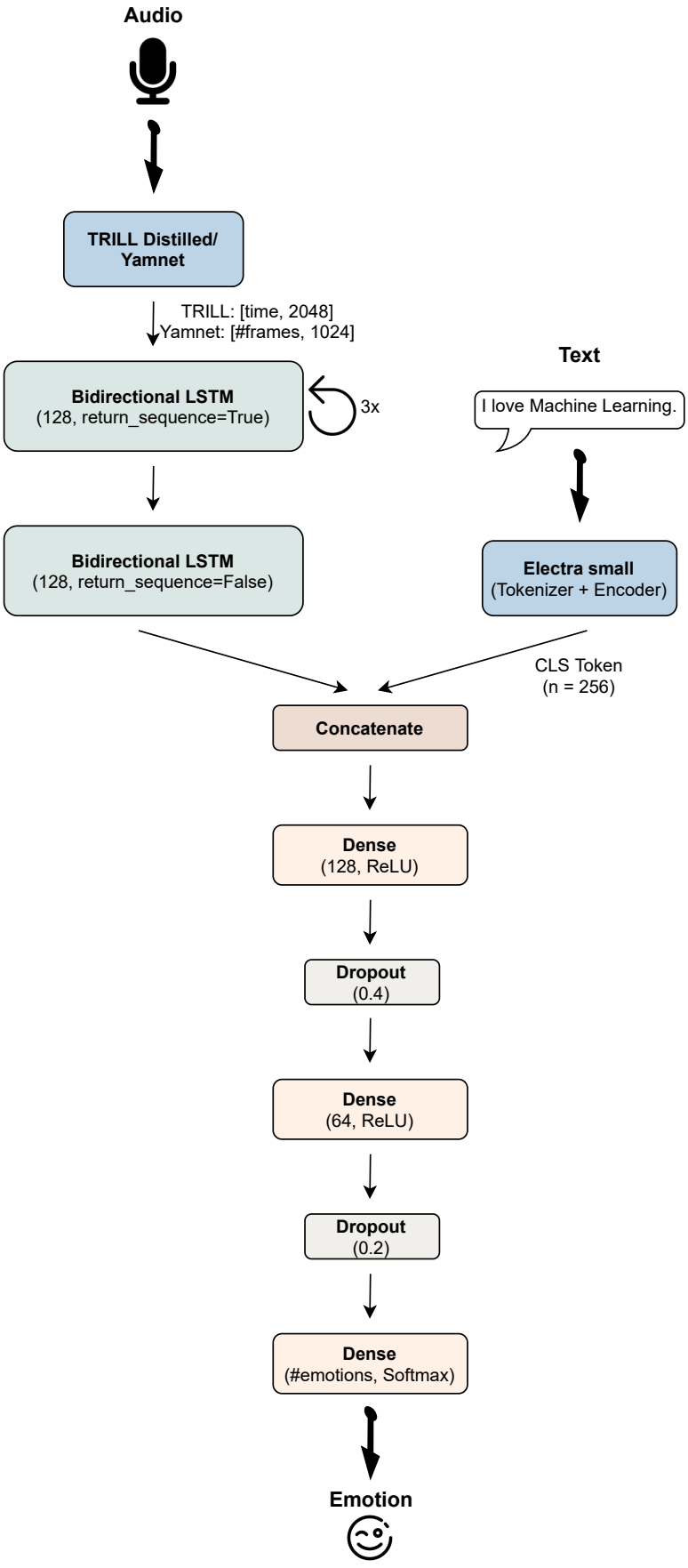

Figure 3.10: A structure of the complex MER model using Electra small, TRILL or YAMNet, LSTM and Dense layers

Chapter 4

Experiments and Results

In this chapter, we are presenting the results of our proposed models and experiments we conducted. The following sections describe each one of them.

4.1 Environment

In this section, we are describing which hardware and software we have used during our work.

4.1.1 Used Hardware

Due to its high time and memory complexity, some of the project's notebooks, especially the ones with the complex models, were trained in Google Colaboratory[1]. Google Colaboratory is a free online Jupyter notebook environment that allows training Machine learning and Deep learning models on CPUs, GPUs, and TPUs[2]. This is very crucial for training Neural Networks because GPUs offers more processing power. So the Deep learning models are trained much faster. Even though Neural Network can be trained on a laptop's GPU, the Google's servers offered in Google Colaboratory have more processing power, which is why we executed the experiments there. The rest of the experiments were executed on our local computers.

4.1.2 Used Software

We developed this project in Python programming language; more specifically, we were programming in Jupyter notebooks[3]. For implementation of the baseline models and TFIDF feature extraction, we used Python's library scikit-learn[4]. The complex models were implemented by using TensorFlow[5]. For reading audio files, we used SoundFile[6] and librosa[7] libraries. Librosa was also used for audio features extraction. For saving trained baseline models, we used pickle[8].

[1] https://colab.research.google.com/
[2] https://www.analyticsvidhya.com/blog/2020/03/google-colab-machine-learning-deep-learning/
[3] https://jupyter.org/
[4] https://scikit-learn.org/stable/
[5] https://www.tensorflow.org/
[6] https://github.com/bastibe/PySoundFile
[7] https://librosa.org/
[8] https://docs.python.org/3/library/pickle.html

We also used Python's libraries numpy[9], pandas[10] and more. We used the free online diagram tool draw.io[11] to create the models' schemas.

4.2 Data preprocessing

We ran all the experiments on IEMOCAP dataset with only 4 basic emotions in total. The basic emotions are **neutral**, **happy**, **sad** and **angry**. We merged happy and excited emotion classes (as it is often done in research papers) to create more balanced classes. The number of utterances for each basic emotion is shown in Table 4.1. We chose not to filter any samples from both PsychExp and RAVDESS datasets, and so we ran the experiments on those complete datasets with all their emotions. We constructed test sets for all the experiments from 20% of all samples. The audio recordings from the IEMOCAP dataset have a variable length, and we needed to have an equal length for all the audio files. That is why we have conducted an analysis and looked at the distribution of the lengths of the audio files from the IEMOCAP dataset. We found out that most of them are short, and only a few samples are significantly longer. Therefore, we decided to use 11 seconds long audio files, which cover 95.74% of all the audio files from the IEMOCAP dataset. The rest of the files were trimmed, and the shorter ones were padded with zeros. We did the same for all the audio files from the RAVDESS dataset, but instead of using 11 seconds, we only used 3 seconds because all of the files were already approximately 3 seconds long.

Table 4.1: A table with number of utterances for each basic emotion in IEMOCAP dataset

Emotion	#utterances
neutral	1708
happy	1636
sad	1084
angry	1103

4.3 Results

4.3.1 Results of TER models

In Table 4.2 are presented results of our TER models used on IEMOCAP dataset and in Table 4.3 are displayed results of the same models on PsychExp dataset. Each model was first trained on the IEMOCAP dataset, and then on the PsychExp Dataset. Then for both datasets, the final accuracy was measured as a mean and standard deviation from 10 iterations over the same data for baseline TER models and Electra small TER model and 5 iterations for the BERT TER model due to its very high time and memory complexity. Figure 4.1 and Figure 4.2 show the confusion matrices of the Elextra small TER model.

[9] `https://numpy.org`
[10] `https://pandas.pydata.org`
[11] `https://app.diagrams.net/`

Table 4.2: A table with results of TER models on IEMOCAP dataset using only basic emotions

Text Emotion Recognition model results on IEMOCAP dataset				
Model name	Train accuracy mean [%]	Train accuracy std [%]	Test accuracy mean [%]	Test accuracy std [%]
LinSVM (TFIDF)	72.54	0.00	59.26	0.00
RF (TFIDF)	68.53	0.34	55.91	0.58
MNB (TFIDF)	72.76	0.00	59.44	0.00
LR (TFIDF)	72.37	0.01	59.19	0.04
BERT	75.92	0.62	**67.08**	0.83
Electra small	87.65	0.54	66.31	1.34

Table 4.3: A table with results of TER models on PsychExp dataset using the full dataset

Text Emotion Recognition model results on PsychExp dataset				
Model name	Train accuracy mean [%]	Train accuracy std [%]	Test accuracy mean [%]	Test accuracy std [%]
LinSVM (TFIDF)	78.04	0.00	57.89	0.00
RF (TFIDF)	75.50	0.26	55.68	0.42
MNB (TFIDF)	80.87	0.00	55.15	0.00
LR (TFIDF)	75.35	0.02	57.69	0.05
BERT	77.94	1.30	**70.41**	0.64
Electra small	80.37	0.50	67.21	0.35

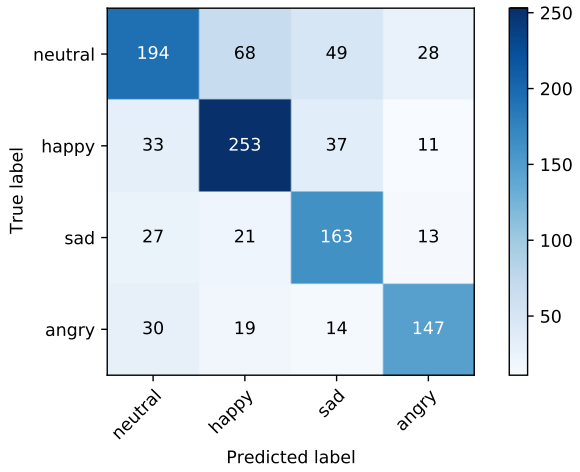

Figure 4.1: A confusion matrix of Electra small TER model trained on IEMOCAP dataset

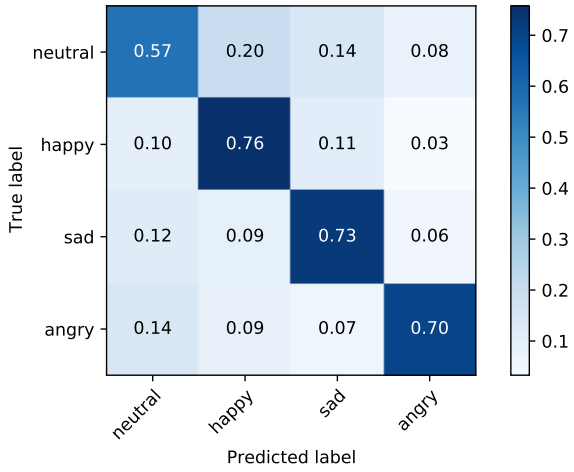

Figure 4.2: A normalized confusion matrix of Electra small TER model trained on IEMOCAP dataset

4.3.2 Selection of audio features for baseline SER models and 1D CNN SER model

In this experiment, our goal was to determine the best possible combination of the basic audio features we have chosen and described before for the baseline SER models and the 1D CNN SER model. For each model, the final test accuracy was computed as a mean of results from 10 iterations, where in each iteration the given model was trained on a 5-fold validation set on the IEMOCAP dataset. Table 4.4 and Table 4.5 show the results of this experiment.

Table 4.4: Ordered maximal test accuracies obtained by the SER models

Feature combination	Test accuracy [%]
mfcc+chroma	59.77
mfcc	59.73
mfcc+mel	58.76
mfcc+mel+chroma	58.71
mel+chroma	55.55
mel	55.11
chroma	39.82

Table 4.5: Ordered mean test accuracies obtained by the SER models

Feature combination	Test accuracy [%]
mfcc+mel+chroma	53.57
mfcc+chroma	53.05
mfcc+mel	53.03
mfcc	53.00
mel+chroma	46.46
mel	44.20
chroma	37.08

4.3.3 Results of Speaker-Independence vs. Speaker-Dependence

In this experiment, we have measured the impact of speaker-dependence vs. speaker-independence on two of our SER models. The speaker-independence means that audio recordings for training, validation, and testing set are from different speakers. On the other hand, in a speaker-dependent setup, the audio recordings of all speakers are randomly split into training, validation, and testing sets, which means that recordings of one speaker can be found in training, validation, and testing sets. This could result in overfitting on some specific speakers. Therefore, the speaker-independent models should be more robust, as [72] states. The Table 4.6 shows the acquired test accuracies for both speaker-dependent and speaker-independent SER models TRILL and YAMNet.

Table 4.6: A table with results of speaker-dependent and speaker-independent SER models

Model name	Speaker-dependent test accuracy	Speaker-independent test accuracy
TRILL	$67.13 \pm 2.32\%$	$50.89 \pm 3.20\%$
YAMNet	$48.39 \pm 1.83\%$	$42.49 \pm 1.32\%$

4.3.4 Results of SER models

In Table 4.7 are presented results of our SER models used on IEMOCAP dataset and in Table 4.8 are displayed results of the same models on RAVDESS dataset. Each model was first trained on the IEMOCAP dataset, and then on the RAVDESS dataset. Then for both datasets, the final accuracies were measured in 3, 5, or 10 iterations over the same data. Figure 4.3 and Figure 4.4 show the confusion matrices of the TRILL SER model. Figure 4.5 and Figure 4.6 show the confusion matrices of the YAMNet SER model.

Table 4.7: A table with results of SER models on IEMOCAP dataset using only basic emotions

Speech Emotion Recognition model results on IEMOCAP dataset				
Model name	Train accuracy mean [%]	Train accuracy std [%]	Test accuracy mean [%]	Test accuracy std [%]
LinSVM	59.58	0.00	59.39	0.05
RF	71.27	0.20	56.31	0.30
MNB	42.11	0.00	42.01	0.00
LR	59.27	0.00	59.53	0.00
1D CNN	61.47	0.65	60.49	0.94
2D CNN	70.84	5.57	58.12	2.89
TRILL	71.35	0.61	**65.19**	0.97
YAMNet	64.92	0.49	60.33	0.78

Table 4.8: A table with results of SER models on RAVDESS dataset using the full dataset

Speech Emotion Recognition model results on RAVDESS dataset				
Model name	Train accuracy mean [%]	Train accuracy std [%]	Test accuracy mean [%]	Test accuracy std [%]
LinSVM	61.42	0.00	48.25	0.00
RF	76.54	0.44	44.79	0.98
MNB	36.31	0.00	33.57	0.00
LR	52.67	0.00	48.25	0.00
1D CNN	71.76	0.90	57.20	1.43
2D CNN	80.18	9.99	57.62	4.94
TRILL	79.46	1.57	**67.13**	2.32
YAMNet	77.03	1.31	48.39	1.83

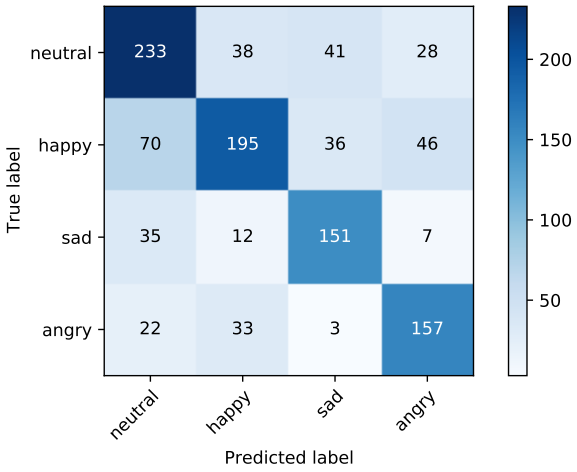

Figure 4.3: A confusion matrix of TRILL SER model trained on IEMOCAP dataset

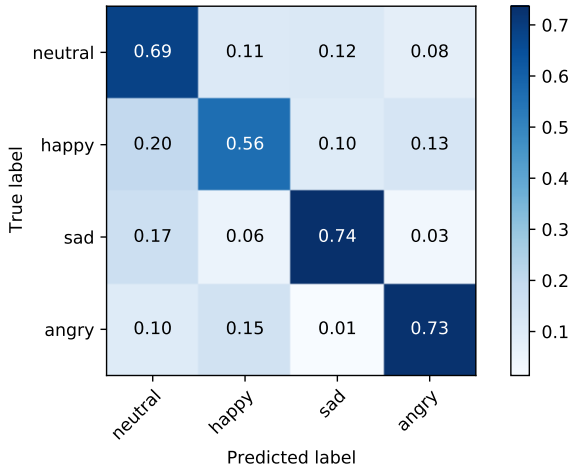

Figure 4.4: A normalized confusion matrix of TRILL SER model trained on IEMOCAP dataset

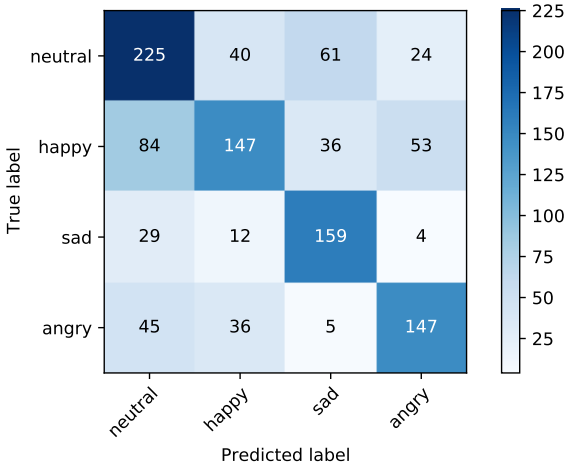

Figure 4.5: A confusion matrix of YAMNet SER model trained on IEMOCAP dataset

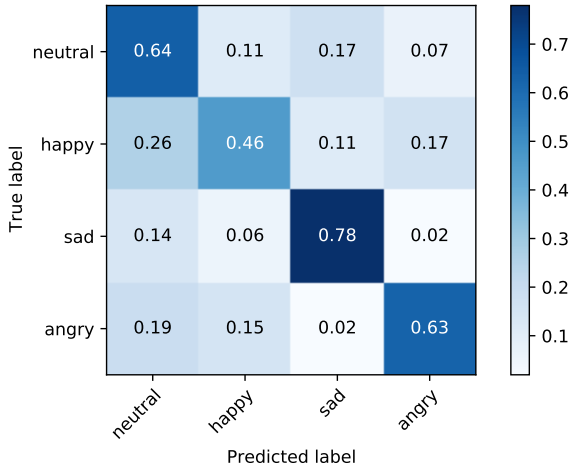

Figure 4.6: A normalized confusion matrix of YAMNet SER model trained on IEMOCAP dataset

4.3.5 Results of MER models

The Table 4.9 shows the results of our baseline and complex MER models used on the IEMOCAP dataset. Each MER model was first trained on the dataset, and then the final accuracies were measured in 5, or 10 iterations over the same data. Figure 4.7 and Figure 4.8 show the confusion matrices of the Electra small with TRILL MER model, and Figure 4.11 shows the training and validation accuracies during the training phase of the model. Figure 4.9 and Figure 4.10 show the confusion matrices of the Electra small with YAMNet MER model, and Figure 4.12 shows the training and validation accuracies during the training phase of the model.

Table 4.9: A table with results of MER baseline models on IEMOCAP dataset using only basic emotions

Multimodal Emotion Recognition baseline model results on IEMOCAP dataset				
Model name	Train accuracy mean [%]	Train accuracy std	Test accuracy mean [%]	Test accuracy std [%]
LinSVM	80.02	0.00	70.37	0.00
RF	73.95	0.16	55.11	0.30
MNB	74.66	0.00	62.69	0.00
LR	80.92	0.00	70.28	0.00
BERT + 1D CNN	85.31	0.71	67.86	1.37
Electra + 1D CNN	85.99	0.71	67.39	0.79
Electra + 2D CNN	78.52	1.01	69.14	1.47
Electra + TRILL	81.15	0.68	**72.81**	0.99
Electra + YAMNet	84.18	0.76	72.14	1.11

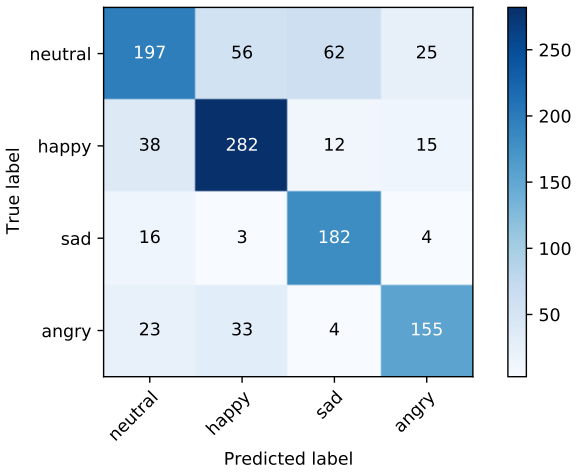

Figure 4.7: A confusion matrix of Electra with TRILL MER model trained on IEMOCAP dataset

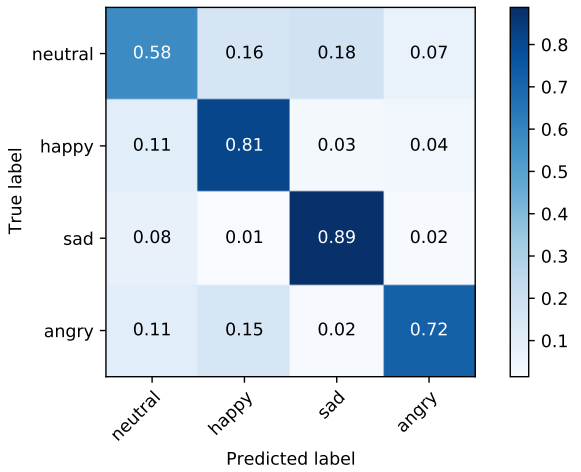

Figure 4.8: A normalized confusion matrix of Electra with TRILL MER model trained on IEMOCAP dataset

Comparison of our MER models with others

We have compared the results of our models with the results of state-of-the-art (SOTA) models [47, 48] and one previous SOTA model [46] on the MER task. The comparison can be found in Table 4.10.

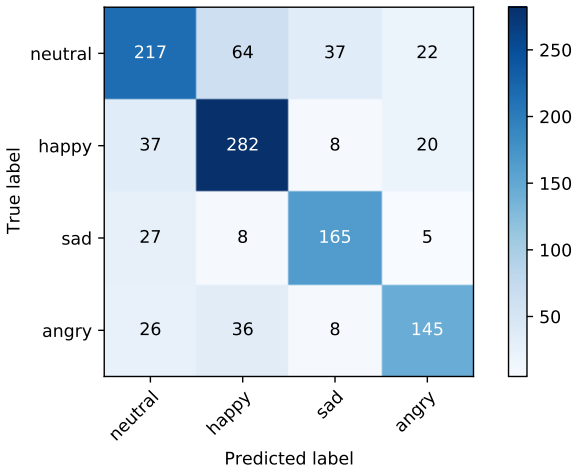

Figure 4.9: A confusion matrix of Electra with YAMNet MER model trained on IEMOCAP dataset

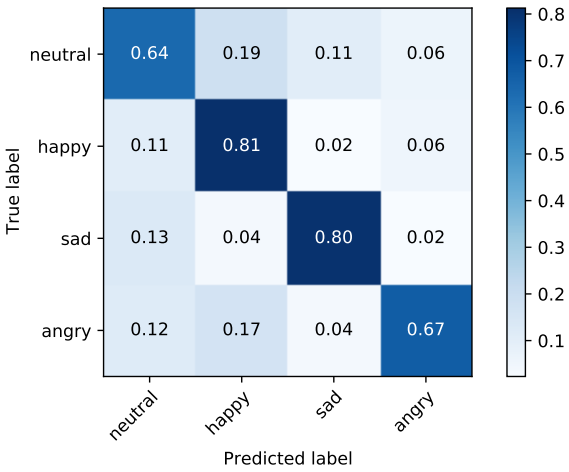

Figure 4.10: A normalized confusion matrix of Electra with YAMNet MER model trained on IEMOCAP dataset

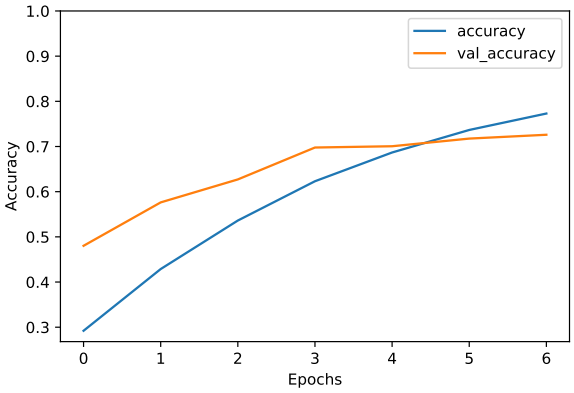

Figure 4.11: Electra with TRILL MER model training graph

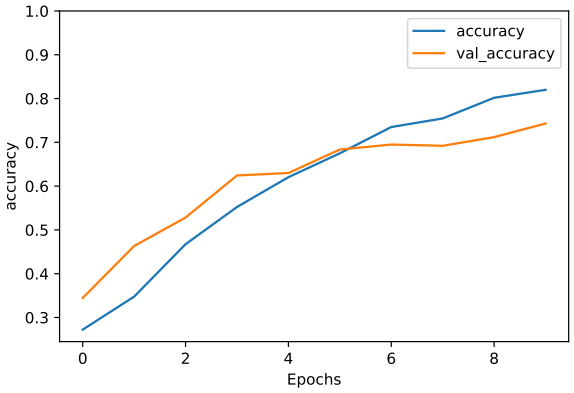

Figure 4.12: Electra with YAMNet MER model training graph

Table 4.10: A table with results of our MER models with related work

Model name	Test accuracy
LinSVM (ours)	$70.37 \pm 0.00\%$
RF (ours)	$55.11 \pm 0.30\%$
MNB (ours)	$62.69 \pm 0.00\%$
LR (ours)	$70.28 \pm 0.00\%$
BERT + 1D CNN (ours)	$67.86 \pm 1.37\%$
Electra + 1D CNN (ours)	$67.39 \pm 0.79\%$
Electra + 2D CNN (ours)	$69.14 \pm 1.47\%$
Electra + TRILL (ours)	$\mathbf{72.81 \pm 0.99\%}$
Electra + YAMNet (ours)	$72.14 \pm 1.11\%$
MDRE [46]	71.80%
Shallow-Fusion of SSL models [47]	75.46%
MHA-2 [48]	**76.50%**

4.4 Discussion

We decided to instead use Electra small than BERT mainly because of the size of the encoders. BERT has 110 million parameters, whereas Electra small has "only" 14 million parameters. To illustrate the difference in the models' sizes, the saved trained BERT TER model was over 1.5GB big. On the other hand, when we saved the Electra TER model into a file, the file was "only" over 150 MB big, so it is a massive difference. The second reason was that the Electra small encoder achieves better results than the BERT small encoder (as is showed in the results of the Electra paper [56]) and is not much worse than the BERT base version we have used.

We decided to choose the combination with all 3 basic audio features (MFCC, Mel Spectrogram, and Chroma) as the best audio feature because it has the highest mean accuracy across all models (Table 4.5). Another option was to select the combination of MFCC and Chroma since it has obtained the highest accuracy (Table 4.4) and has the second-best mean accuracy across all models.

As we explained the difference between speaker-independence and speaker-dependence, it is clear that all SER models should be speaker-independent so that they are more robust and possibly not overfitted on some speakers. Despite that, we decided to create speaker-dependent models because we have found only one paper [72] that was referring to this problem. Therefore, we presumed that all the other papers had not considered this. So in order to be able to compare the results of our work with the previous work, we had to make this decision. Furthermore, in the IEMOCAP dataset, we could not find which actors recorded which files, so there was no way to ensure that we could build speaker-independent on the IEMOCAP dataset.

We expected the final test accuracies of the speaker-independent SER models to be lower than the results of the same but speaker-dependent models. The reason for that is because the speaker-independent task is much more challenging. After all, the models are being tested against speakers they have not heard before. Therefore, they cannot memorize some distinctive features in the speakers' voices. The difference between the speaker-independent and speaker-dependent TRILL SER model is quite significant, which was quite surprising for us. Interestingly, the speaker-independent TRILL SER model was still able to outperform all speaker-dependent baseline SER models and even the speaker-dependent version of the YAMNet SER model. This shows that the TRILL model is quite suited for the SER task.

We decided to use a slightly shorter classification layer for the Electra with 1D CNN MER model because the model with the shorter classification architecture had achieved slightly better results than the longer one used in all the other MER models.

The baseline models performed relatively similarly in the TER task, but the worst baseline model was the Random Forrest. The BERT TER model performed the best out of all our proposed TER models. However, the Electra small TER model was not far behind it, especially on the IEMOCAP dataset, even though that the Electra small model has much fewer parameters. In the SER task, by far, the worst result got the Multinomial Naive Bayes model. The Logistic Regression and Linear Support Vector Machines models performed the best out of our

baseline models in the SER task and the MER task. This shows that both of these models are pretty suited for the task of Emotion Recognition regardless of the modality they use (text, or speech, or both). On the other hand, the MNB model proved that it is well suited for the TER task. The RF model showed that it is not very much suited for any Emotion Recognition task. We have also noticed that the LinSVM and LR MER models have even beaten some of our complex MER models and that they are not that far behind our best-performing proposed MER models while being much simpler and faster because of their simplicity. Therefore, they should also be considered when deciding which MER model(s) to use. In the SER task, the best performing model of all our proposed SER models was the TRILL SER model. We think that the YAMNet SER model did not perform so well, especially on the RAVDESS dataset, because it has very sophisticated architecture (inspired by the MobileNet v1 architecture) and requires a vast dataset, which, unfortunately, RAVDESS does not seem to be even though it is not a tiny dataset. It just seems that it is not enough for the YAMNet SER model, but it does not need to be true. It could also mean that the model could be over sophisticated. In the MER task, the two best-performing models are the Electra small with TRILL and Electra small with YAMNet models. Even though the difference between them and the current SOTA solutions is not that small, they were able to beat the previous SOTA solution MDRE [46] from 2018, which we consider an achievement.

Chapter 5

Practical Applications

In this chapter, we are presenting two of our demo practical applications of Emotion Recognition models. Both of these applications are available via web browser, and each has a short description in the following sections.

5.1 Web Application with an API for TER

5.1.1 Description

The web application consists of two parts, one is a GUI for entering texts for evaluating the emotional state of the given text, and the other one is a REST API [74], which can be used for emotion evaluation from other applications. The MNB TER model is used in this application. The model was trained on the PsychExp dataset using TFIDF vector features. The Figure 5.1 shows the application's GUI. The application is available at `http://ter-simple.herokuapp.com/`.

5.1.2 Used software

This web application was developed in Flask[1], which is one of Python's web development libraries. The web application was deployed using Heroku[2] platform. The implementation was inspired by Flask web forms tutorial[3]. We followed steps from the Deploying a Flask Application to Heroku tutorial[4] during the deployment of the page.

[1] `https://flask.palletsprojects.com/en/1.1.x/`
[2] `https://www.heroku.com`
[3] `https://pythonspot.com/flask-web-forms/`
[4] `https://stackabuse.com/deploying-a-flask-application-to-heroku/`

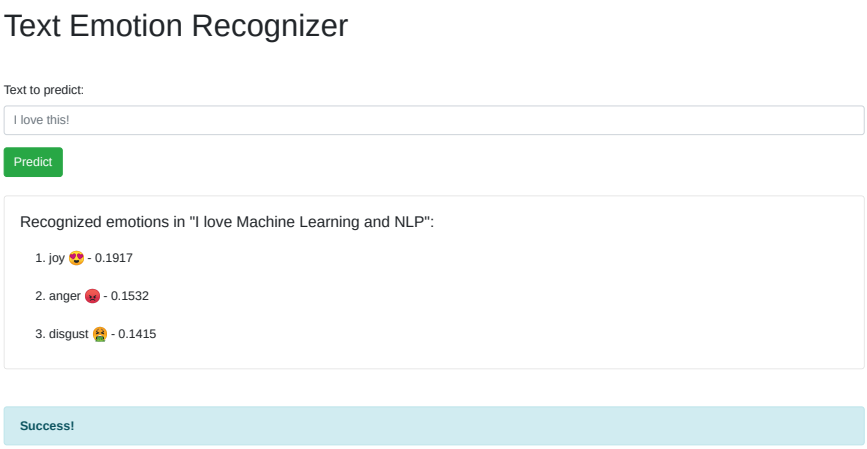

Figure 5.1: A showcase of the GUI of the web application for Text Emotion Recognition (TER)

5.2 Demo on Google Colaboratory

5.2.1 Description

This demo is available at this url[5]. It is in a Jupyter notebook on Google Colaboratory that is available via web browser from anywhere. The only thing that the user has to do is to run the code cells. Then by either recording a speech or uploading audio files in *wav* format, the audio is then passed into a transcriptor to get an audio transcript. Then, the audio and transcript are passed into several TER, SER, and MER models to get a prediction. We have used Electra small with TRILL and Electra small with YAMNet MER models for audio and text, Electra small TER model for text, and TRILL and YAMNet SER models for audio.

5.2.2 Used software

We used a code for recording audio from the Direct access to your webcam and microphone inside Google Colab notebook post[6] and code for uploading file from the Load local data files to Colaboratory post[7] on Stack Overflow. For transcription we have used a Python's library SpeechRecognition[8]. The larger trained models are loaded from Amazon Simple Storage Service (Amazon S3)[9].

[5] https://colab.research.google.com/github/HonzaCuhel/mer-thesis-app/blob/main/predict_emotion_mer_thesis_app.ipynb
[6] https://ricardodeazambuja.com/deep_learning/2019/03/09/audio_and_video_google_colab/
[7] https://stackoverflow.com/questions/47320052/load-local-data-files-to-colaboratory
[8] https://github.com/Uberi/speech_recognition
[9] https://aws.amazon.com/s3/

Chapter 6

Conclusion

6.1 Summary of project

In this thesis, we have proposed several models for Emotion Recognition tasks from text, audio, or a combination of both. They return an emotion that best suits the given input based on the models' knowledge obtained during their training. We have presented four baseline models and then some complex models and compare their performance with each other and with results of some state-of-the-art solutions. The results were pleasantly surprising, and we think that we got relatively good results. Linear SVM and Logistic Regression models overall did best out of our baseline models. They showed that they are both suited for the Emotion Recognition task from either text, audio, or both inputs. Both of these models acquired test accuracy in the MER task over 70%, which is an excellent result compared to the related work. Multinomial Naive Bayes performed well in the TER task. It has shown us that it is suited for the TER task, but on the other hand, it is not suited for the SER task. The BERT TER model performed the best in the TER task, and the TRILL SER model performed best in the SER task. The best-performing MER models were the Electra small with TRILL and Electra small with Yamnet, which were even able to beat the previous state-of-the-art solution MDRE [46] from 2018. We have also created two practical demo applications for us and others to try out some of our trained models.

6.2 Future Work

There is certainly room for improvement. Some areas which could be improved are the following: create more complex models, experiment more with the speaker-independence or experiment with different datasets; for example, for MER, the CSU-MOSEI could be utilized as it seems promising because it is larger than the IEMOCAP dataset, or use other modalities like video, or apply audio augmenting techniques to make the models more robust, or try Emotion Recognition with a Dimensional emotion model.

Bibliography

1. CALVO, Rafael A.; MAC KIM, Sunghwan. EMOTIONS IN TEXT: DIMENSIONAL AND CATEGORICAL MODELS. *Computational Intelligence*. 2013, vol. 29, no. 3, pp. 527–543. Available from DOI: https://doi.org/10.1111/j.1467-8640.2012.00456.x.
2. ACHEAMPONG, Francisca Adoma; WENYU, Chen; NUNOO-MENSAH, Henry. Text-based emotion detection: Advances, challenges, and opportunities. *Engineering Reports*. 2020, vol. 2, no. 7, e12189. Available from DOI: https://doi.org/10.1002/eng2.12189.
3. ALSWAIDAN, N., MENAI, M.E.B. A survey of state-of-the-art approaches for emotion recognition in text. *Knowledge and Information Systems*. 2020, vol. 62, no. 8, 2937–2987. Available from DOI: https://doi.org/10.1007/s10115-020-01449-0.
4. SAILUNAZ, K., DHALIWAL, M., ROKNE, J. ET AL. Emotion detection from text and speech: a survey. *Social Network Analysis and Mining*. 2018, vol. 8, no. 1, p. 28. Available from DOI: https://doi.org/10.1007/s13278-018-0505-2.
5. EKMAN, Paul. Basic emotions. *Handbook of cognition and emotion*. 1999, vol. 98, no. 45-60, p. 16.
6. PLUTCHIK, Robert. A general psychoevolutionary theory of emotion. In: *Theories of emotion*. Elsevier, 1980, pp. 3–33.
7. RUSSELL, James A. A circumplex model of affect. *Journal of personality and social psychology*. 1980, vol. 39, no. 6, p. 1161.
8. SCARANTINO, Andrea; SOUSA, Ronald de. Emotion. In: ZALTA, Edward N. (ed.). *The Stanford Encyclopedia of Philosophy* [https://plato.stanford.edu/entries/emotion/]. Summer 2021. Metaphysics Research Lab, Stanford University, 2021.
9. *Swiss Center For Affective Sciences Swiss Center for Affective Sciences*. 2019. Available also from: https://www.unige.ch/cisa/research/materials-and-online-research/research-material/.
10. *PsychExp*. 2017. Available also from: https://github.com/bfelbo/DeepMoji/tree/master/data/PsychExp.
11. MOHAMMAD, Saif; BRAVO-MARQUEZ, Felipe; SALAMEH, Mohammad; KIRITCHENKO, Svetlana. Semeval-2018 task 1: Affect in tweets. In: *Proceedings of the 12th international workshop on semantic evaluation*. 2018, pp. 1–17.

12. CHEN, Sheng-Yeh; HSU, Chao-Chun; KUO, Chuan-Chun; HUANG, Ting-Hao K.; KU, Lun-Wei. EmotionLines: An Emotion Corpus of Multi-Party Conversations. *CoRR*. 2018, vol. abs/1802.08379. Available from arXiv: 1802.08379.

13. DEMSZKY, Dorottya; MOVSHOVITZ-ATTIAS, Dana; KO, Jeongwoo; COWEN, Alan; NEMADE, Gaurav; RAVI, Sujith. *GoEmotions: A Dataset of Fine-Grained Emotions*. 2020. Available from arXiv: 2005.00547 [cs.CL].

14. LI, Yanran; SU, Hui; SHEN, Xiaoyu; LI, Wenjie; CAO, Ziqiang; NIU, Shuzi. DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, 2017, pp. 986–995. Available also from: https://www.aclweb.org/anthology/I17-1099.

15. ALM, Ebba Cecilia Ovesdotter. *Affect data (distributed by Cecilia Ovesdotter Alm)*. [N.d.]. Available also from: http://people.rc.rit.edu/~coagla/affectdata/index.html.

16. AMAN, Saima. *Emotion-Annotated Dataset Aman*. [N.d.]. Available also from: http://saimacs.github.io/.

17. BUECHEL, Sven; HAHN, Udo. Emobank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. 2017, pp. 578–585.

18. *Sentiment Analysis in Text - dataset by crowdflower*. 2016. Available also from: https://data.world/crowdflower/sentiment-analysis-in-text.

19. WALLACH, Hanna M. Topic modeling: beyond bag-of-words. In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 977–984.

20. JONES, Karen Sparck. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*. 1972.

21. MIKOLOV, Tomas; CHEN, Kai; CORRADO, Greg; DEAN, Jeffrey. *Efficient Estimation of Word Representations in Vector Space*. 2013. Available from arXiv: 1301.3781 [cs.CL].

22. PENNINGTON, Jeffrey; SOCHER, Richard; MANNING, Christopher D. Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

23. BOJANOWSKI, Piotr; GRAVE, Edouard; JOULIN, Armand; MIKOLOV, Tomas. *Enriching Word Vectors with Subword Information*. 2017. Available from arXiv: 1607.04606 [cs.CL].

24. RADFORD, Alec; WU, Jeffrey; CHILD, Rewon; LUAN, David; AMODEI, Dario; SUTSKEVER, Ilya. Language models are unsupervised multitask learners. *OpenAI blog*. 2019, vol. 1, no. 8, p. 9.

25. DEVLIN, Jacob; CHANG, Ming-Wei; LEE, Kenton; TOUTANOVA, Kristina. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*. 2018, vol. abs/1810.04805. Available from arXiv: 1810.04805.

26. HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. *Neural computation*. 1997, vol. 9, no. 8, pp. 1735–1780.

27. FELBO, Bjarke; MISLOVE, Alan; SØGAARD, Anders; RAHWAN, Iyad; LEHMANN, Sune. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017. Available from DOI: 10.18653/v1/d17-1169.

28. LIVINGSTONE, Steven R.; RUSSO, Frank A. *The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)*. Zenodo, 2018. Version 1.0.0. Available from DOI: 10.5281/zenodo.1188976. Funding Information Natural Sciences and Engineering Research Council of Canada: 2012-341583 Hear the world research chair in music and emotional speech from Phonak.

29. DUPUIS, Kate; PICHORA-FULLER, M Kathleen. Toronto emotional speech set (TESS)-Younger talker_Happy. 2010.

30. *Surrey Audio-Visual Expressed Emotion (SAVEE) Database*. [N.d.]. Available also from: http://kahlan.eps.surrey.ac.uk/savee/.

31. CAO, Houwei; COOPER, David G; KEUTMANN, Michael K; GUR, Ruben C; NENKOVA, Ani; VERMA, Ragini. Crema-d: Crowd-sourced emotional multimodal actors dataset. *IEEE transactions on affective computing*. 2014, vol. 5, no. 4, pp. 377–390.

32. BURKHARDT, Felix; WEISS, Benjamin; KIENAST, Miriam; PAESCHKE, Astrid. *Berlin Database of Emotional Speech*. [N.d.]. Available also from: http://emodb.bilderbar.info/docu/.

33. MCKEOWN, Gary; VALSTAR, Michel F; COWIE, Roderick; PANTIC, Maja. The SEMAINE corpus of emotionally coloured character interactions. In: *2010 IEEE International Conference on Multimedia and Expo*. 2010, pp. 1079–1084.

34. PEETERS, Geoffroy. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. *CUIDADO Ist Project Report*. 2004, vol. 54, no. 0, pp. 1–25.

35. COHN, Richard. Introduction to neo-riemannian theory: a survey and a historical perspective. *Journal of Music Theory*. 1998, pp. 167–180.

36. PAUWS, Steffen. Musical key extraction from audio. In: *ISMIR*. 2004.

37. VENKATARAMANAN, Kannan; RAJAMOHAN, Haresh Rengaraj. Emotion Recognition from Speech. *CoRR*. 2019, vol. abs/1912.10458. Available from arXiv: 1912.10458.

38. DAVIS, Steven; MERMELSTEIN, Paul. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*. 1980, vol. 28, no. 4, pp. 357–366.

39. SCHNEIDER, Steffen; BAEVSKI, Alexei; COLLOBERT, Ronan; AULI, Michael. wav2vec: Unsupervised Pre-training for Speech Recognition. *CoRR*. 2019, vol. abs/1904.05862. Available from arXiv: 1904.05862.

40. SHOR, Joel; JANSEN, Aren; MAOR, Ronnie; LANG, Oran; TUVAL, Omry; QUITRY, Felix de Chaumont; TAGLIASACCHI, Marco; SHAVITT, Ira; EMANUEL, Dotan; HAVIV, Yinnon. Towards learning a universal non-semantic representation of speech. *arXiv preprint arXiv:2002.12764*. 2020.

41. SAMANTARAY, A. K.; MAHAPATRA, K.; KABI, B.; ROUTRAY, A. A novel approach of speech emotion recognition with prosody, quality and derived features using SVM classifier for a class of North-Eastern Languages. In: *2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS)*. 2015, pp. 372–377. Available from DOI: 10.1109/ReTIS.2015.7232907.

42. ZHAO, Jianfeng; MAO, Xia; CHEN, Lijiang. Speech emotion recognition using deep 1D & 2D CNN LSTM networks. *Biomedical Signal Processing and Control*. 2019, vol. 47, pp. 312 –323. ISSN 1746-8094. Available from DOI: https://doi.org/10.1016/j.bspc.2018.08.035.

43. C. BUSSO, M. BULUT, C.C. LEE, A. KAZEMZADEH, E. MOWER, S. KIM, J.N. CHANG, S. LEE, AND S.S. NARAYANAN. IEMOCAP: Interactive emotional dyadic motion capture database. *Journal of Language Resources and Evaluation*. 2008, vol. 42, no. 4, pp. 335–359. Available from DOI: https://doi.org/10.1007/s10579-008-9076-6.

44. BAGHER ZADEH, AmirAli; LIANG, Paul Pu; PORIA, Soujanya; CAMBRIA, Erik; MORENCY, Louis-Philippe. Multimodal Language Analysis in the Wild: CMU-MOSEI Dataset and Interpretable Dynamic Fusion Graph. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 2236–2246. Available from DOI: 10.18653/v1/P18-1208.

45. PORIA, Soujanya; HAZARIKA, Devamanyu; MAJUMDER, Navonil; NAIK, Gautam; CAMBRIA, Erik; MIHALCEA, Rada. MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversations. *CoRR*. 2018, vol. abs/1810.02508. Available from arXiv: 1810.02508.

46. YOON, Seunghyun; BYUN, Seokhyun; JUNG, Kyomin. *Multimodal Speech Emotion Recognition Using Audio and Text*. 2018. Available from arXiv: 1810.04635 [cs.CL].

47. SIRIWARDHANA, Shamane; REIS, Andrew; WEERASEKERA, Rivindu; NANAYAKKARA, Suranga. *Jointly Fine-Tuning "BERT-like" Self Supervised Models to Improve Multimodal Speech Emotion Recognition*. 2020. Available from arXiv: 2008.06682 [eess.AS].

48. YOON, Seunghyun; BYUN, Seokhyun; DEY, Subhadeep; JUNG, Kyomin. Speech Emotion Recognition Using Multi-hop Attention Mechanism. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019. ISBN 9781479981311. Available from DOI: 10.1109/icassp.2019.8683483.

49. SAHU, Gaurav. Multimodal Speech Emotion Recognition and Ambiguity Resolution. *CoRR*. 2019, vol. abs/1904.06022. Available from arXiv: 1904.06022.

50. CORTES, Corinna; VAPNIK, Vladimir. Support-vector networks. *Machine learning*. 1995, vol. 20, no. 3, pp. 273–297.

51. HO, Tin Kam. Random decision forests. In: *Proceedings of 3rd international conference on document analysis and recognition*. 1995, vol. 1, pp. 278–282.

52. BREIMAN, Leo. Random forests. *Machine learning*. 2001, vol. 45, no. 1, pp. 5–32.

53. QUINLAN, J. Ross. Induction of decision trees. *Machine learning*. 1986, vol. 1, no. 1, pp. 81–106.

54. KIBRIYA, Ashraf M; FRANK, Eibe; PFAHRINGER, Bernhard; HOLMES, Geoffrey. Multinomial naive bayes for text categorization revisited. In: *Australasian Joint Conference on Artificial Intelligence*. 2004, pp. 488–499.

55. HOSMER JR, David W; LEMESHOW, Stanley; STURDIVANT, Rodney X. *Applied logistic regression*. John Wiley & Sons, 2013.

56. CLARK, Kevin; LUONG, Minh-Thang; LE, Quoc V; MANNING, Christopher D. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*. 2020.

57. VASWANI, Ashish; SHAZEER, Noam; PARMAR, Niki; USZKOREIT, Jakob; JONES, Llion; GOMEZ, Aidan N.; KAISER, Lukasz; POLOSUKHIN, Illia. Attention Is All You Need. *CoRR*. 2017, vol. abs/1706.03762. Available from arXiv: 1706.03762.

58. *Classify text with BERT*. TensorFlow, [n.d.]. Available also from: https://www.tensorflow.org/tutorials/text/classify_text_with_bert.

59. DESARDA, Akash. *Working with Hugging Face Transformers and TF 2.0*. Towards Data Science, 2020. Available also from: https://towardsdatascience.com/working-with-hugging-face-transformers-and-tf-2-0-89bf35e3555a.

60. SRIVASTAVA, Nitish; HINTON, Geoffrey; KRIZHEVSKY, Alex; SUTSKEVER, Ilya; SALAKHUTDINOV, Ruslan. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*. 2014, vol. 15, no. 1, pp. 1929–1958.

61. PLAKAL, Manoj; ELLIS, Dan. *YAMNet*. [N.d.]. Available also from: https://github.com/tensorflow/models/tree/master/research/audioset/yamnet.

62. *Mel Frequency Cepstral Coefficient (MFCC) tutorial*. [N.d.]. Available also from: http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/.

63. FLANAGAN, James L. *Speech analysis synthesis and perception*. Springer Science & Business Media, 2013.

64. PANDEY, Prateek Kumar; GUPTA, Anurag; WADHWA, Mohit. *Speech Emotion Recognition (SER) through Machine Learning*. 2021. Available also from: `https://www.analyticsinsight.net/speech-emotion-recognition-ser-through-machine-learning/`.

65. GEMMEKE, Jort F; ELLIS, Daniel PW; FREEDMAN, Dylan; JANSEN, Aren; LAWRENCE, Wade; MOORE, R Channing; PLAKAL, Manoj; RITTER, Marvin. Audio set: An ontology and human-labeled dataset for audio events. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017, pp. 776–780.

66. JANSEN, Aren; PLAKAL, Manoj; PANDYA, Ratheet; ELLIS, Daniel P. W.; HERSHEY, Shawn; LIU, Jiayang; MOORE, R. Channing; SAUROUS, Rif A. Unsupervised Learning of Semantic Audio Representations. *CoRR*. 2017, vol. abs/1711.02209. Available from arXiv: `1711.02209`.

67. HERSHEY, Shawn; CHAUDHURI, Sourish; ELLIS, Daniel PW; GEMMEKE, Jort F; JANSEN, Aren; MOORE, R Channing; PLAKAL, Manoj; PLATT, Devin; SAUROUS, Rif A; SEYBOLD, Bryan, et al. CNN architectures for large-scale audio classification. In: *2017 ieee international conference on acoustics, speech and signal processing (icassp)*. 2017, pp. 131–135.

68. HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

69. HOWARD, Andrew G; ZHU, Menglong; CHEN, Bo; KALENICHENKO, Dmitry; WANG, Weijun; WEYAND, Tobias; ANDREETTO, Marco; ADAM, Hartwig. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*. 2017.

70. ROCKIKZ, Abdou. *How to Make a Speech Emotion Recognizer Using Python And Scikit-learn*. 2019. Available also from: `https://www.thepythoncode.com/article/building-a-speech-emotion-recognizer-using-sklearn`.

71. FUKUSHIMA, Kunihiko; MIYAKE, Sei. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In: *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285.

72. VENKATARAMANAN, Kannan; RAJAMOHAN, Haresh Rengaraj. Emotion Recognition from Speech. *CoRR*. 2019, vol. abs/1912.10458. Available from arXiv: `1912.10458`.

73. EBERSBACH, Mike; HERMS, Robert; EIBL, Maximilian. Fusion Methods for ICD10 Code Classification of Death Certificates in Multilingual Corpora. In: *CLEF (Working Notes)*. 2017.

74. FIELDING, Roy T. *Architectural styles and the design of network-based software architectures*. University of California, Irvine Irvine, 2000.