Real-time Speech Emotion Recognition (RSER) in Wireless Multimedia Sensor Networks

by

Serik Zhilibayev

Submitted to the School of Engineering and Digital Sciences
in partial fulfillment of the requirements for the degree of

Master of Data Science

at the

NAZARBAYEV UNIVERSITY

July 2021

Author . . . .
School of Engineering and Digital Sciences
29 July, 2021

Certified by . . . .
Adnan Yazici
Full Professor
Thesis Supervisor

Certified by . . . .
Enver Ever
Associate Professor
Thesis Co-supervisor

Accepted by . . . .
Vassilios D. Tourassis
Dean, School of Engineering and Digital Sciences

Real-time Speech Emotion Recognition (RSER) in Wireless Multimedia Sensor Networks

by

Serik Zhilibayev

Submitted to the School of Engineering and Digital Sciences
on 29 July, 2021, in partial fulfillment of the
requirements for the degree of
Master of Data Science

Abstract

Recently Wireless Multimedia Sensor Networks (WMSN) is extensively used and huge amounts of data are generated on daily basis. There are huge processes that have to be monitored in real-time, so preprocessing and fast analysis of raw data is required to be done and stored on the edge. Since, edge computation allows the environment to be decentralized, which makes it highly responsive, low price, scalable, and secure. WMSN and edge computing are important in areas like healthcare where the subject has to be monitored and analyzed continuously. In this work, we propose the healthcare system for monitoring human emotion using speech in realtime (RSER). Firstly, this project aims to analyze state-of-the are SER approaches with respect to time and the ability to work on constrained devices. Secondly, the new approach based on time analysis will be provided. There will be Exploratory data Analysis on multiple datasets that will be used for training such as the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) , Berlin (EMO-DB) and IMEOCAP datasets. Data based on Vocal tract spectrum features and low-level acoustic features (Pitch and energy) will be extracted. The data will be trained and evaluated on Deep Learning and Machine Learning algorithms. Algorithms will be prioritized by their time, energy, and accuracy metrics. Then, this experiment will be tested and evaluated on embedded device (Raspberry PI). Finally, modified model based on algorithm analysis will be tested on 3 Scenarios (Processing on Edge, Processing Sink, and Streaming).

Thesis Supervisor: Adnan Yazici
Title: Full Professor

Thesis Co-supervisor: Enver Ever
Title: Associate Professor

Acknowledgments

I would like to thank my adviser Adnan Yazici for enormous help in conducting this research, for providing all resources and being excellent mentor.

Also would like to thank my co-adviser Enver Ever for giving me valuable feedbacks and for always being ready to help us.

I also would like to thank my friends and lab-mates Serik Almakhan and Chingizkhan Tangirbergen for emotional help and for establishing warm environment in the lab.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human emotions are one of the significant indicators of the state of mind and health. With instant mental and physical deviations, a person cannot consciously understand the seriousness of the situation. This means that he needs an assistant who can control his emotions. So, there is huge attention to exploration in this field. Recently, researchers use speech to identify human emotions. Consequently, there is big progress in Speech Emotion Recognition. The interesting thing is that, a human sound can be represented in various ways. So this rich variety helps to find correlations with human emotions. There are a number of quality techniques were proposed by scientists in terms of accuracy. However, these algorithms are expensive in terms of speed and hardware implementation. Since humans can't effort themselves high-powered devices and bear them on a daily basis. Therefore, this field needs more investigation into speed and the capability of work in constrained devices.

Wireless Multimedia Sensor Network (WMSN) is widely used in various spheres such as healthcare, agriculture, and etc. It generates huge amounts of various data from environmental sensors and multimedia devices such as cameras and microphones. Multimedia data is very heavy and the raw transition can drastically overwhelm the throughput of the network. So, the needed context has to be extracted in order to minimize and clean the data. Therefore, edge computation allows the environment to be decentralized, which makes it highly responsive, low price, scalable, and secure. Therefore, current embedded systems provide enough computational power to perform Real-time Speech Emotion Recognition.

There are 3 aims of this project. Firstly, to analyze the time complexity and ability to work on edge devices of Deep Learning approaches. Secondly, based on results from the previous step develop RSER system. Also, to test and analyse proposed SER Model in 3 WMSN scenarios.

In this paper, the research on RSER will be proposed. Firstly, a brief review of the literature related to Speech Emotion Recognition will be presented. Secondly, the description of datasets, feature extraction, and methodology, and results will be demonstrated.

1.1 Related Works

Deep Learning architectures such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) are widely used in solving the speech-emotion recognition problem [1, 2, 6]. CNN helps to find concealed features, and RNN finds special sequence-dependent patterns from the audio file. And the consolidation of these architectures builds up a model that classifies speech emotions [1, 2]. In fact, one of the biggest challenges is to find a feature that significantly emphasizes the behavior of speech at different emotions. There are many ideas that researchers come up with. In one of the researches raw audio was used by Trigeorgis et al. [8] as an input. There CNN model was used to suppress noise. And Recurrent Long-Short Term Memory architecture was trained to classify speech emotions. The proposed model outperformed Support Vector Regressors that were trained on two different sets of acoustic parameters of an audio such as eGeMAPS [8] and ComParE [10].

Tarantino et al. [7] introduced CNN based self-attention architecture and windowing techniques that used eGeMAPS features to classify emotions. They assert that RNN models suffer from decaying memory, which makes them unable to preserve correlations. On the other hand, self-attention remembers all correlations and is able to make less computations in classification. In fact, results showed that CNN model that was trained on acoustic features is better than RNN model trained on raw audio. In addition, Triantafyllopoulos et al. [16] used the same technique combining eGeMAPS [8] and ComParE [10] feature sets. This approach significantly improved results.

Transformation of raw audio to 2D representation was proposed by Lim et al. [1]. Researchers used short-time Fourier transform (STFT) to create an image from an EMO-DB [4] dataset audio. The image was fed into the model. The model was based on CNN and LSTM architectures. CNNs trained to extract features and LSTM architectures were learned to recognize sequential dynamics. So, they called their architecture “Time distributed CNN”.

The approach mentioned above gave a start on applying various acoustic 2-D representations of audio. According to EMO-DB dataset [4], the best results are shown by Zhao et al. [5] and Demircan and Kahramanli [11]. Firstly, Zhao et al. [5] trained Time distributed CNN model on Log-Mel spectrograms and provided classification with 95.89% accuracy. Secondly, Demircan and Kahramanli [11] trained classical machine learning algorithms with Mel Frequency Cepstral coefficients (MFCCs). Then, they applied a fuzzy C-means clustering dimensionality reduction technique. So, the best result was demonstrated by support vector machines (SVM) and k-nearest neighbors (kNN) with test accuracies 92.86% and 92.86%, respectively.

Zhang et al. [12] combined speech and song samples from RAVDESS [3] dataset. They proposed a multi-task classification approach. 4 classificators were trained in (“Speech”, “Male”), (“Speech”, “Female”), (“Song”, “Male”), (“Song”, “Female”) manner. Then, they applied 5 different decision-making algorithms using results from 4 classifiers such as Decision trees, Majority vote, etc. Unfortunately, they obtained an accuracy of 57.14%.

Like-wise Zeng et al. [13] also combined speech and song samples from RAVDESS [3] dataset. However, researchers propose Gated Residual Networks (GResNet) which allows training multi-task classifiers jointly. They firstly extract spectrograms from the audio, then they feed it to the introduced Deep Neural Network. Results are significantly higher than models that were trained in a single task manner, which is 71% accuracy.

Issa et.al [14] implemented deep CNN architecture that trains on multiple frequency domain features (MFCC, Chromogram, Mel spectrogram, Contrast, Tonnetz). This research was extensively investigated. For example, the proposed model was trained on RAVDESS[3] and EMO-DB [4] datasets and demonstrated high testing accuracies in each dataset: 71.61%, 86.1%, and 64.03% respectively. So, they outperformed previous state-of-the-art approaches.

Previously mentioned works show how accuracy significantly improved by advanced techniques in recent years. However, there are few works that demonstrate the speed and complexity of SER algorithms. Also, there are no related works that show how quality SER algorithms run on WMSN embedded devices such as Raspberry pi, etc. Fortunately, there is research that is similar to ours. Silva et. al [15] conducted research on Urban Sound Classification in WMSN. They evaluated ML algorithms on embedded devices with respect to accuracy and execution time. They compared the performances between powered and constrained devices. So, they show that quality stays the same, but speed performance is 10 times lower on embedded systems. Also, they mention that the feature extraction from the audio is the most time-consuming part. In order to improve speed performance on constrained devices, they suggest decreasing the size of the input audio frame.

Chapter 2

Databases

For the thesis 3 datasets are used. Each dataset contains its own specific feature. RAVDESS dataset is recorded from large number of speakers and it has significant number of speech samples. And, EMODB dataset contains non-english data. Also, TESS dataset contains large number utterances only with two speakers. Each of the database has potentially significant impact on finding common emotional features from speakers with the diverse backgrounds.

2.1 RAVDESS

The first dataset that was chosen is the Ryerson Audio-Visual Database of Emotional Speech and Song [3]. And there are eight different emotion classes such as calm, neutral, happy, angry, disgust, fearful, surprised and sad. The dataset was collected from 24 actors where 12 are males and 12 female. They were recorded audio and video of face by pronouncing English sentences. Only audio recordings will be used for this research. As a result, the total number of utterances is 1440. The Figure 2-1 depicts distribution and amplitude information of the database. Each emotion contains approximately 190 utterances, but only neutral class contains 90 utterance. Average duration of audio samples is approximately 3.5 seconds. And, each sample of audio is padded with nearly 0.75 seconds of silence.

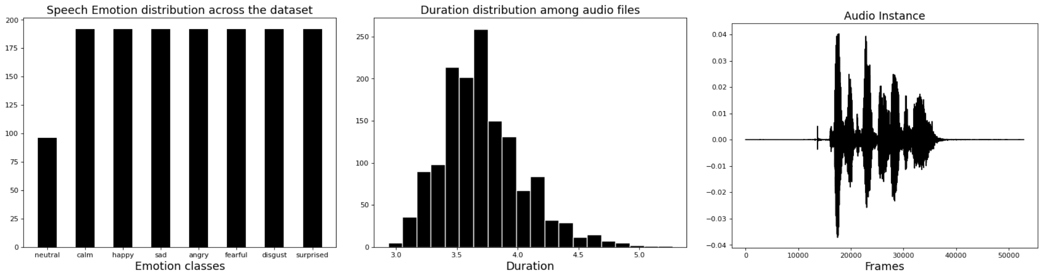


Figure 2-1: RAVDESS Dataset Information

2.2 EMODB

The second dataset was taken from Berlin Database of Emotional Speech [16] (EMODB). It is open German speech database that contains audios with seven emotions: happiness, sadness, anger, fear, disgust, boredom, and neutral. Speech samples were recorded by 5 male and 5 female people and each subject produced 10 utterances for each emotion. Utterances were recorded with sampling rate 48Hz. Totally it contains 535 audio files. The Figure 2-2 depicts distribution and amplitude information of the database. Each emotion contains on average 80 utterances. However, 'angry' emotion contains 120 speech samples. and disgust class has only 50 speech samples. Overall, this database is small and distributed not evenly. Also, average duration of audio samples is approximately 3.5 seconds. However, the duration of audios are not distributed normally, it contains few audios that are more than 6 seconds. And, samples of audio are not padded with silence.

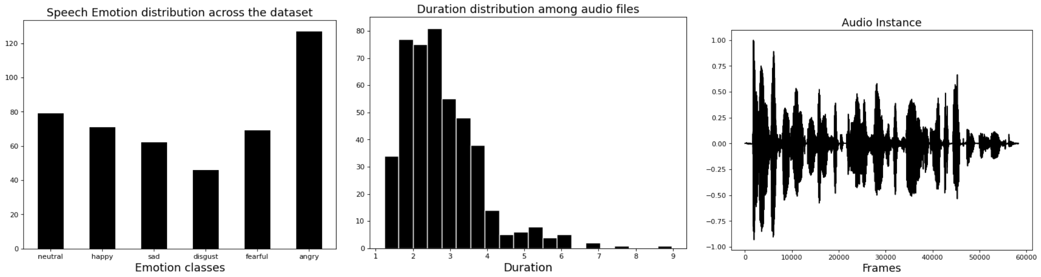


Figure 2-2: EMODB Dataset Information

2.3 IMEOCAP

EMEOCAP dataset [18] was also employed. This database contains a set of 200 target words spoken by two English actors aged 26 and 64. It depicts audios with seven emotions (happiness, anger, disgust, fear, pleasant surprise, neutral and sadness) and overall it contains 2800 audio files. Utterances were recorded with sampling rate 48Hz. IMEOCAP involves only two speakers. The Figure 2-3 depicts distribution and amplitude information of the database. Each emotion contains equally 400 utterances. Average duration of audio samples is approximately 2 seconds. Durations of speech samples are normally distributed. And, samples of audio are not padded with silence.

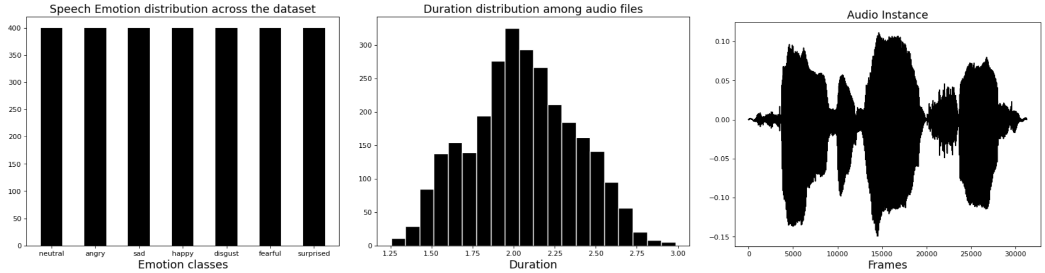


Figure 2-3: IMEOCAP Dataset Information

Chapter 3

Methodology

3.1 Digital Sound

In physics the sound is product of vibrations and collisions of gas molecules entering to the human ears by varying air pressure. The vibration of air molecules from the source will cause vibration of surrounding particles by chain reaction creating sound wave. In order to record the sound, it have to be converted to digital sound. The incoming data captured by microphone converting variations of air pressure to variations of voltages. Then, analog electric signals converted to digitized version using Analog-to-Digital-Converter (ADC). Signal by itself continuous data stream. So, during digitization process sound is collected as discrete data at defined sampling rate in time domain and magnitude is quantized at a defined bit-depth. Mostly, quality sound recorded with 44100 Hz sampling rate and 16 bit bit-depth. Therefore, digital sound is one-dimensional sequence of numbers. Usually, uncompressed digital sounds are stored in WAV PCM format. Recorded sounds can have multiple channels (for example Stereo), but for Machine Learning purpose single channel audio (Mono) is used.

3.2 Preprocessing

To transform the audio in a proper format data preprocessing operations are used. It includes data cleaning, transformation and augmentation. It helps to handle irrelevant and missing data samples. Given datasets does not contain missing or irrelavent data, but RAVDESS dataset have to be transformed to have common format. Also, EMODB dataset contains insufficient data, so some data enrichment techiniques have to be used.

3.2.1 Silence Removal

As I mentioned before RAVDESS dataset's audio samples are padded with 0.75 seconds of silence from both starting and ending of a audio file. This kind of data might be sufficient to be trained for models that only using one dataset. However, models trained on padded audios may negatively effect on prediction of RSER model. Since, Voice Activity Algorithms that segment speeches in audio stream will produce data samples with no silence padding. Consequently, there might be a misunderstanding between incoming speech samples and a model trained on padded utterances.

To remove silent paddings energy of every frame is used. Energy of an audio sample is measured in Decibel (dB). For each chunk of the audio with length 0.2 seconds the mean power is calculated.

$$\mu_{Power} = \sum S^2/n$$

Then, with the given equation we can calculate the energy of the chunk. The lower bound threshold for the energy is taken as 40 dB .

$$Energy = 10 * \log_{10}(\mu_{Power})$$

3.2.2 Data Augmentation

Deep Learning models significantly improve by using huge amount of data because they can find more patterns to find differences between classes and similarities within the class. So, by increasing the amount of the training data models can avoid overfitting issues and find a way to significantly generalize. Mining new data samples is expensive so Data Augmentation techniques are used to deal with deficiency of data. Data Augmentation is technique to increase data by using data itself. This method can be applied to any kind of data such as numeric, acoustic and image data. Usually, generated sample is similar to original data but affected by synthetic transformations. For example, images can be rotated and signals can have noise.

For the thesis 3 data augmentation techniques are applied for EMODB dataset (fig). Since, the dataset have only 535 data samples.

- **Adding Noise** - Adding noise to the training set can give a significant regularization effect on the learning process. For the each sample of the audio there were added a noise with +15 Db signal to noise ratio (SNR). The SNR metric helps to not overbalance noise effect on original audio. The SNR is defined as $10log10(P_{speech}/P_{noise})$, where P is the mean power of the audio signal.

- **Shift the Audio** - it shift the audio to the left or to the right in the time domain by some amount of seconds.

- **Stretch** - This technique changing the speed in two ways such as make faster or make slower.

The samples to be augmented are choosen randomly. The amount of the augmented data is predefined.

3.3 Acoustic Features

Feature of the speech is a transformation of the data from wave form to parametric representation for processing and analysis purposes. Features help to understand the data in multi-dimensional spectrum. For example, the data that was recorded has time domain information. Consequently, using SFFT we can convert the data to be in frequency and time domain. So, by changing the representation of the sound wave more information can be learned about it. For instance, there are more features can be extracted from audio data. There are two mostly used feature classes in SER are prosody and spectral features.

- **Prosodic features** - these features analyse the sound in long term basis such as long connected speech. Specifically, it shows behaviour of the intonation, rhythm and stress. They are also called supersegmental features because it needs minimum 30-100ms of sound duration for analysis [10]. Mostly duration, intensity, fundamental frequency (pitch) and long-term spectral features are used to extract prosodic features. For example, statistical values (minimum, maximum, mean, standard deviation, etc) derived from time domain amplitudes or from fundamental frequencies are called prosodic features. The advantage of them is well distinguishing between low and high arousal emotions (happy and bored). However, disadvantage is poor classification between the same arousal emotions (sad and bored).

- **Spectral features** - they are also called segmental features because they extract features in short time periods such as 10-30 ms] [10]. The aim of the spectral analysis is to extract the energy information at different frequency levels in the short time period. These features are good at modelling vocal fold vibration. Possible disadvantage is that they poorly performs on speaker independent emotion recognition. The most prevalent spectral feature in recent researches is Mel Frequency Cepstral Coefficients.

In this research 6 feature extraction methods will be analysed:

1. Chroma: Chroma features are known as pitch class profiles. They show to what pitch class current sound is related. It is widely used in identifying correlation between human timber and musical aspect of harmony because of its robustness.

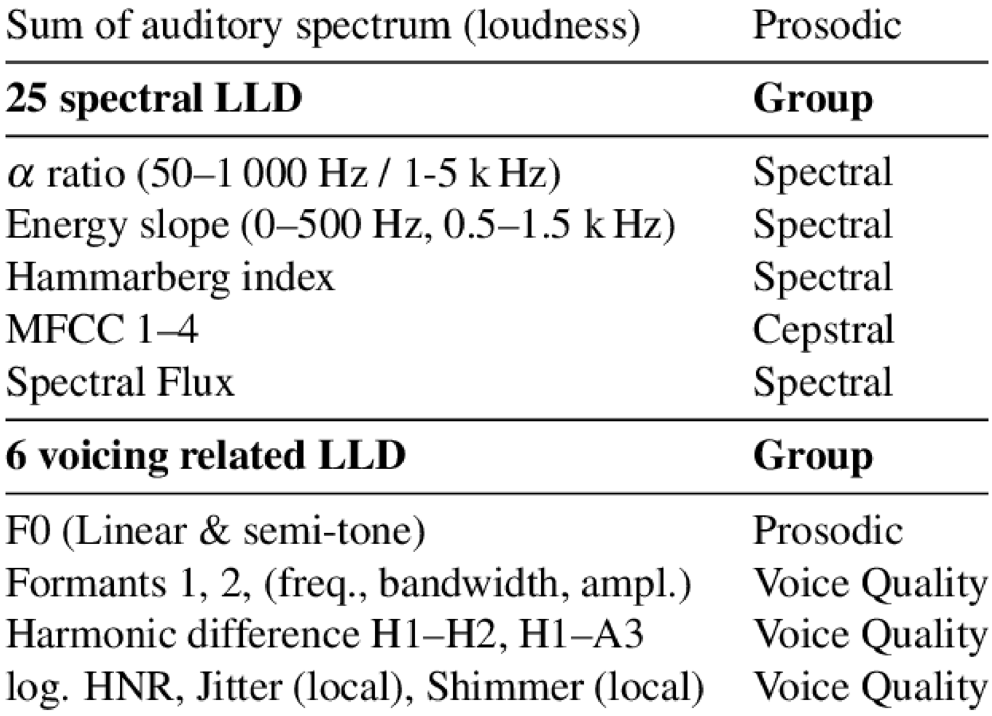
2. Spectral Contrast: Spectral representation of the sound may significantly suffer from high noise level. To overcome this kind of issue Spectral Contrast features are used. Spectral contrast evaluates difference of frequency levels between picks and valleys. So, these features can help to enhance the intelligibility of the sound in high SNR environment.

3. Tonnetz: Tonnetz is used to detect changes in fundamental frequency (Harmonic wave). As Chroma features detect the class of the pitch, Tonnetz features identify pitch behaviour in temporal dimension.

4. Mel-Spectrogram: As it was mentioned before Mel Spectrogram is a representation of the signal strength in frequency and time domain. This is fundamental basis for extracting spectral features mentioned above. Also, Mel-spectrograms are used in learning CNN models to classify acoustic sounds. Since, it gives an opportunity to use sound as a picture. Firstly, to obtain Mel-Spectrogram a spectrogram is derived with Short Term Fourier Transform. Then, the resulted spectrogram is transformed to human perceptual scale (Mel Scale).

5. MFCC: MFCCs is a inverse Fourier Transform of Mel-Spectrum. Human sounds are results of the air pressure passed through vocal tract (including teeth, tongue etc). The envolope of the time power spectrum of the speech signal is characterization of the vocal tract. So, MFCC accurately represents this vocal shape. They are widely applied in Deep Learning models such as Speech Recognition and Text-to-Speech jobs.

6. Common Standards (LLD): There are many acoustic features to extract from the sound. Researchers from fields related to Speech Analysis and Speech recognition created standard set of feature extraction methods. For this thesis two standards are analysed. First, extended Geneva Minimalistic Acoustic Parameter Set (eGeMAPS) which contains 42 low-level-descriptors [b7]. Second, Interspeech Computational Paralinguistics Challenge features set which contains 65 low-level-descriptors [b7].



Sum of auditory spectrum (loudness)	Prosodic
25 spectral LLD	**Group**
α ratio (50–1 000 Hz / 1-5 k Hz)	Spectral
Energy slope (0–500 Hz, 0.5–1.5 k Hz)	Spectral
Hammarberg index	Spectral
MFCC 1–4	Cepstral
Spectral Flux	Spectral
6 voicing related LLD	**Group**
F0 (Linear & semi-tone)	Prosodic
Formants 1, 2, (freq., bandwidth, ampl.)	Voice Quality
Harmonic difference H1–H2, H1–A3	Voice Quality
log. HNR, Jitter (local), Shimmer (local)	Voice Quality

Figure 3-1: eGeMAPS feature set

3.4 Voice Activity Detection Algorithm

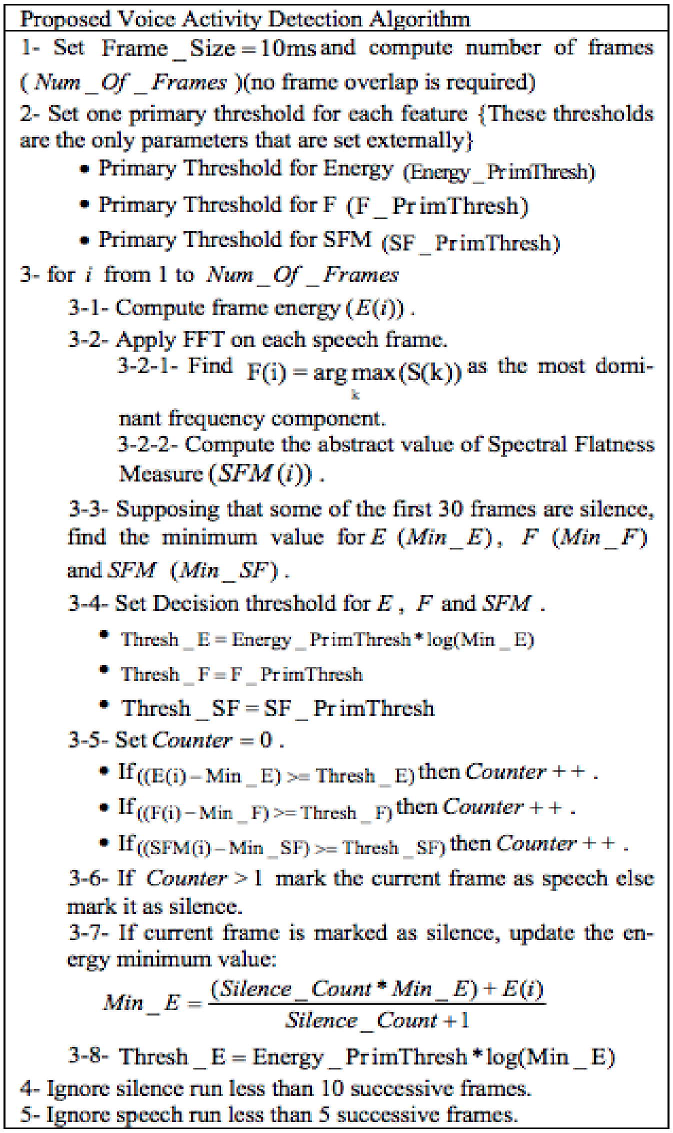
Voice activity detection is a algorithm that segments speech regions from the audio stream. It is an important front end preprocessing step for Real-time Speech emotion recognition applications. The performance of VAD algorithms is significantly impacts Speech emotion recognition results. For the purpose of examining real-time speech emotion recognition on embedded devices the VAD algorithm have to be simple and robust in noisy environment. So, for the thesis an algorithm proposed by Moattar et.al [18] is utilized. In their research they propose three features to identify speech parts of the sound:

- **Spectral flatness** provides a way to quantify how tone-like a sound is, as opposed to being noise-like. The probability that a sound is a noise.

- **Short Term Energy** it is measured with Decibels (dB). Perceptible loudness indicator.

- **Most dominant Frequency Component** - Frequency level with maximum magnitude.

The proposed algorithm starts with framing the audio stream chunk. Initial N frames are used for threshold precomputation. For the following frames three metrics are evaluated. If one or more conditions are satisfied with thresholds, given frame is labeled as a speech frame. In the below figure detailed pseudo implementation can be seen.



```
Proposed Voice Activity Detection Algorithm
1- Set $Frame\_Size = 10ms$ and compute number of frames ($Num\_Of\_Frames$)(no frame overlap is required)
2- Set one primary threshold for each feature {These thresholds are the only parameters that are set externally}
        • Primary Threshold for Energy $(Energy\_PrimThresh)$
        • Primary Threshold for F $(F\_PrimThresh)$
        • Primary Threshold for SFM $(SF\_PrimThresh)$
3- for $i$ from 1 to $Num\_Of\_Frames$
    3-1- Compute frame energy $(E(i))$.
    3-2- Apply FFT on each speech frame.
        3-2-1- Find $F(i) = \arg\max_k(S(k))$ as the most dominant frequency component.
        3-2-2- Compute the abstract value of Spectral Flatness Measure $(SFM(i))$.
    3-3- Supposing that some of the first 30 frames are silence, find the minimum value for $E$ $(Min\_E)$, $F$ $(Min\_F)$ and $SFM$ $(Min\_SF)$.
    3-4- Set Decision threshold for $E$, $F$ and $SFM$.
        • $Thresh\_E = Energy\_PrimThresh * \log(Min\_E)$
        • $Thresh\_F = F\_PrimThresh$
        • $Thresh\_SF = SF\_PrimThresh$
    3-5- Set $Counter = 0$.
        • If $((E(i) - Min\_E) >= Thresh\_E)$ then $Counter++$.
        • If $((F(i) - Min\_F) >= Thresh\_F)$ then $Counter++$.
        • If $((SFM(i) - Min\_SF) >= Thresh\_SF)$ then $Counter++$.
    3-6- If $Counter > 1$ mark the current frame as speech else mark it as silence.
    3-7- If current frame is marked as silence, update the energy minimum value:
        $Min\_E = \frac{(Silence\_Count * Min\_E) + E(i)}{Silence\_Count + 1}$
    3-8- $Thresh\_E = Energy\_PrimThresh * \log(Min\_E)$
4- Ignore silence run less than 10 successive frames.
5- Ignore speech run less than 5 successive frames.
```

Figure 3-2: VAD Algorithm

3.5 Speech Recognition Models

In this research 4 types of Deep Learning models are experimented for testing accuracy, time and energy consumption. This models proposed by 4 research papers. So, they are reimplemented and tested on embedded device. Below detailed information about architectures is provided.

3.5.1 1D CNN Model

Issa et al.[14] provided a 1D CNN architecture that consumes 1 dimensional input array. Input is stacked array of mean intensity of MFCC ,spectral-contrast, Mel-Spectrogram, chroma, tonnetz at each frequency level. The base-line CNN architecture constructed from 1D convolution layer followed by dropout, batch normalization and activation layers (ReLU).

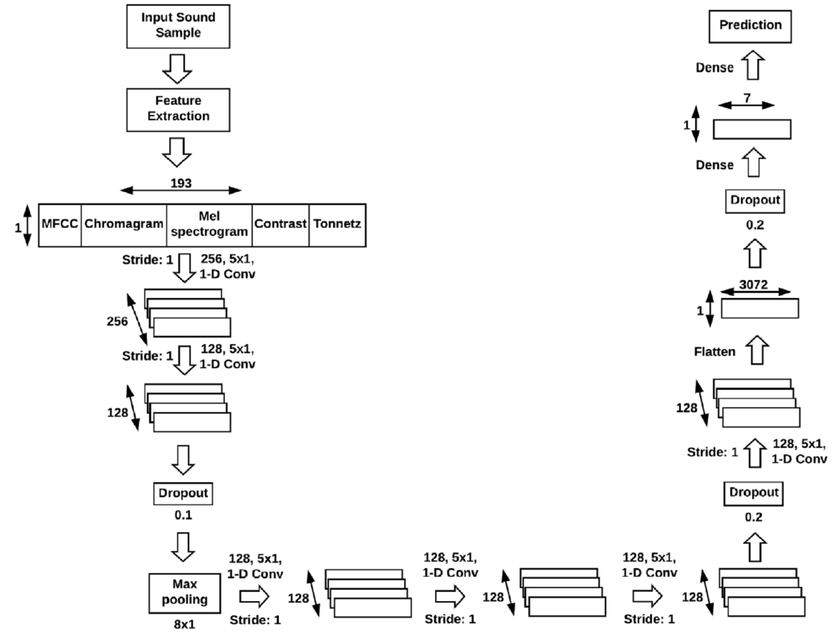


Figure 3-3: 1D CNN model Architecture

3.5.2 2D CNN model Architecture

To learn speech samples as an images 2D CNN model is used. The model is trained on Mel-spectrograms. Spectrograms are derived by STFT with Hamming window of 5ms and 4.4 ms overlap. The frequency information above 4KHz is removed. In result, 129 frequency levels are represented in sepctrograms. The spectrogram is resized to have 129x129 shape and fed to the network as an image with one channel.2D CNN Network is constructed with two 2D Convolution layers followed by Max Pooling layers. After flattening the feature space, one hidden dense layer is followed.

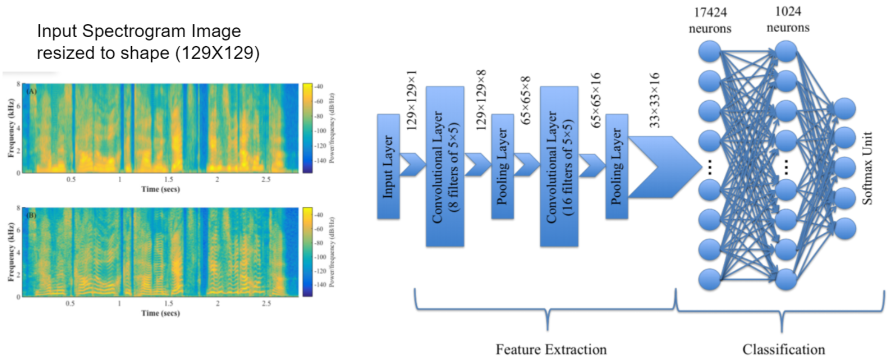


Figure 3-4: as

3.5.3 1D CNN LSTM Model

To examine speech emotion recognition without using any acoustic feature extraction LSTM model is implemented. To build this model the architecture introduced by Zhao et al[11] is used. This model consumes raw audio as an input. The input size is 64000 frames (4 seconds with sampling rate 16K Hz) of audio. The model is starting with four 1D convolution layers followed by Batch Normalization, Activation and MaxPooling Layers. After which sequenced with LSTM layer. Finally, LSTM layer connects to fully connected layer which contains weights for each emotion class.

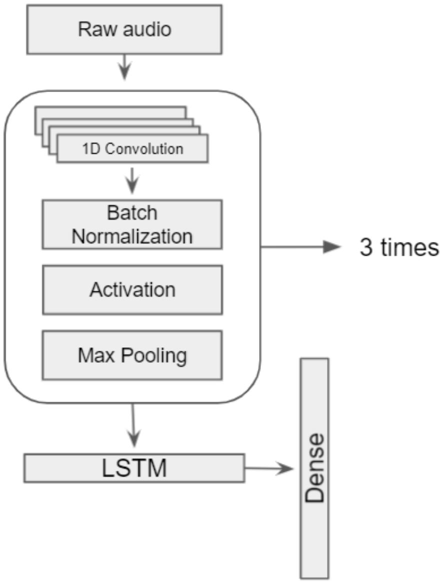


Figure 3-5: LSTM Model

3.5.4 Self-Attention Model using 1D CNN

Self-Attention mechanism is used to lower computation complexity on learning sequential data. Self-Attention models became superior to RNN models due to its simplicity and performance. Tarantino et el [7] provided Self-Attention model to predict Speech emotions using eGeMAPS feature set as an input. The architecture starts with 6 sequentially connected 1D Convolution Layer and Max-Pooling layer pairs. Then, the last Max-Pooling layer connects with Self-Attention Layer. Finally, Self-Attention layer is followed by 2D Convolution, Max-Pooling and Dense layer.

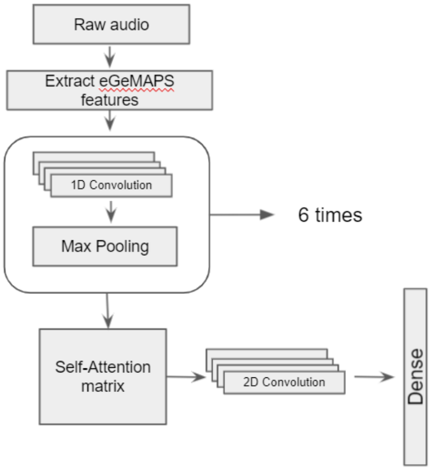


Figure 3-6: Self-Attention model

3.6 Hardware Setup

Two hardware devices are used:

3.6.1 Raspberry Pi

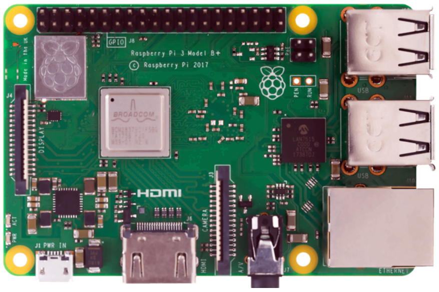


Figure 3-7: Raspberry pi 3B+

Raspberry Pi is a small sized pocket computer which is called micro-controller. This device is used to run read incoming audio stream, run VAD and SER algorithms on real-time basis. Its main features are:

- Model: Raspberry Pi 3B+
- SoC: Broadcom BCM2837

- CPU: 4× ARM Cortex-A53, 1.2GHz
- GPU: Broadcom VideoCore IV
- RAM: 1GB LPDDR2 (900 MHz)
- Networking: 10/100 Ethernet, 2.4GHz 802.11n wireless
- 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet

3.6.2 Voltage Tester

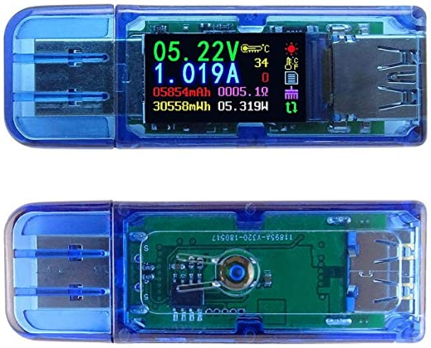


Figure 3-8: USB Voltage Tester RuiDeng AT34

Voltage Tester is a device that detects voltage, current, power consumption in real-time. This device is used to measure energy consumption of Speech Emotion Recognition in real-time. This device is connected between Raspberry pi and power source.

- Voltage measurement range: 3.70-30.00V
- Voltage measurement resolution: 0.01V
- Current measurement range: 0-4.000A
- Current measurement resolution: 0.001A
- Capacity accumulation range: 0-99999mAh

- Energy accumulation range: 0-99999mWh-999.99Wh
- Power measurement range: 0-120W
- Temperature range:0-80℃
- Temperature measurement error: ±3℃

3.6.3 Edge TPU USB Accelerator

Figure 3-9: Coral TPU Accelerator

Coral Edge TPU is a coprocessor which accelerates Tensorflow Deep Learning computations. It is widely used with constrained devices like Raspberry Pi to enhance computation power used for running complex DL models. In this project TPU accelerator is used to check whether it can enhance SER models computation speed.

- Google Edge TPU coprocessor: 4 TOPS (int8); 2 TOPS per watt
- USB 3.0 Type-C* (data/power)

3.7 Real-time Prediction Scenarios

There are 3 main scenarios are implemented. There are three ways how can we run RSER. Firstly, running VAD and SER on edge and then sending results to sink.

Secondly, running VAD on the edge, send speech samples to the sink and predict emotions on sink. Finally, stream audio to sink from edge, and run VAD and SER on sink.

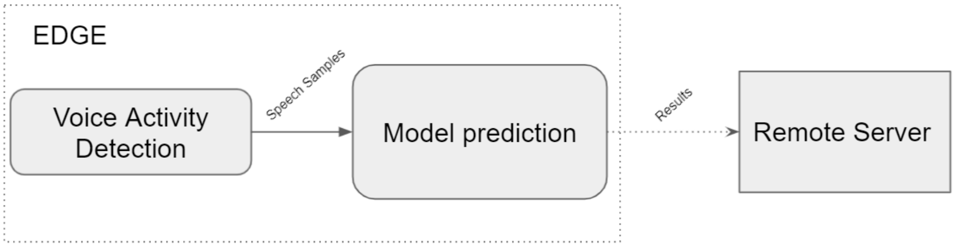


Figure 3-10: Full RSER on Edge

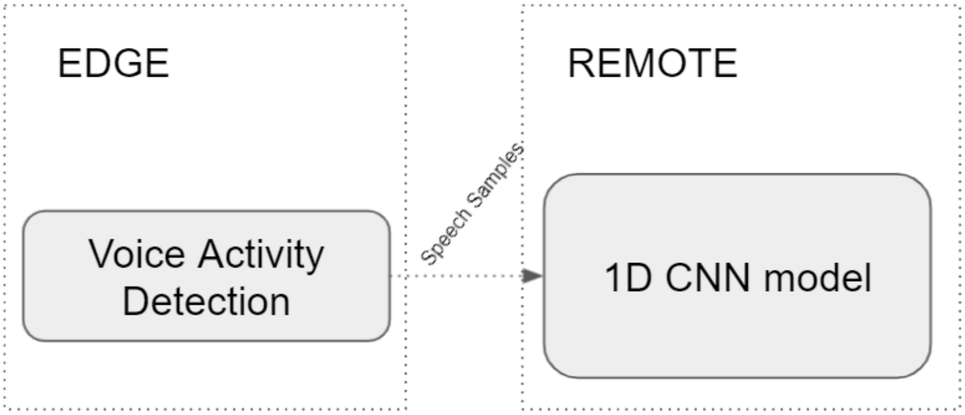


Figure 3-11: Local VAD and remote SER

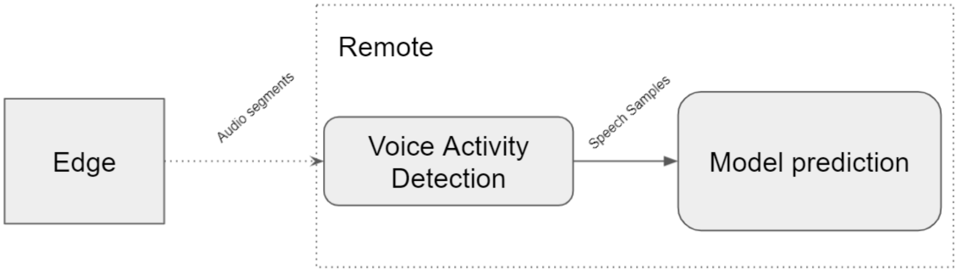


Figure 3-12: Stream to Remote and SER remotely

Chapter 4

Results and Discussion

As it was mentioned before Real-time Speech Emotion recognition models have to be tested in all steps starting from feature extraction, SER, and VAD step. All steps will include speed, accuracy, and energy consumption results run on Raspberry Pi.

4.1 Feature Extraction Results

Table 4.1: Spectral Features Speed Test

Feature	Size	RPi (s)
stft	1025	0.083
mfccs	40	0.214
chroma	12	0.261
mel	128	0.185
contrast	7	0.078
tonnetz	6	6.09

Feature Extraction is the main basis for establishing SER on embedded devices. There are two types of features spectral features and LLD sets. 100 audio samples with average of 3 seconds duration were used to extract features. Here in Table 4.1

can be seen that most of the spectral features are computed in less than half second. The shortest time taken for spectral contrast (0.078 s). But the longest computation is held by tonnetz 6.09 seconds. The reason why tonnetz takes so long to be computed is the precomputation of two consecutive operations. They are extraction of stft and getting harmonic elements from stft output.

Table 4.2: Low Level Desciptors Speed Test

Low Level Descripters	Size	RPi (s)
egemaps	88	0.2078
compare	6373	0.5691
gemaps	62	0.3752

Low Level Descriptors are also analyzed for this research. Same 100 audio samples were used to extract LLDs. The results are can be seen in Table 4.2. Here we can see that eGeMAPS is the lightest way of LLD extraction (0.2078 s). And, ComparE is the slowest (0.5691 s). Even GeMAPS LLDs has the smallest size, it is a bit slower than eGeMAPS extraction. It might be that parameters of feature extraction methods and implementations of feature extraction methods are different.

4.2 Modelling results

As it was mentioned before, 4 different SER models are analysed in this research. These models are analysed in terms of accuracy and time consumption.

4.2.1 1D CNN model

Table 4.3: 1D CNN Model

Feature Sets	Input Size	RAVDESS	EMODB	IMEOCAP
All 5 Spectral Features	193	71.30%	86.10%	64.30%
MFCC, CHROMA, MelSpectrogram, Contrast	187	69.30%	82.86%	65.40%
Egemaps	88	59.00%	50.78%	51%
Compare	6373	55.00%	49.65%	54%
Gemaps	62	52.00%	51.56%	49%
All features + Egemaps	281	65.00%	50.50%	60%

1D CNN model that was provided by Dias et al. was performed approximately the same as it was claimed in their research. The original model that uses 5 spectral features (Table 4.1) achieved 71.3%, 86.1%, and 64.3% on RAVDESS, EMODB, IMEOCAP respectively. However, this model contains the slowest features extraction method (Tonnetz). So this model was also analysed without tonnetz feature. The results obtained from the modified feature set performed a little bit poor but yet they are good 69.3%, 82.86% and 65.4% on RAVDESS, EMODB, IMEOCAP respectively. This model was also trained the standard feature sets such as eGeMAPS, ComparE and GeMAPS. However, the results were significantly lower than the previous two models varying between 50 - 55 % on each dataset.

Table 4.4: 1D model Speed Test

Model Inference Speed	Only inference	Spectral Features without tonnetz
Raspberry Pi	0.047	0.768 s
PC	0.013	0.066 s

For testing time consumption the second model is chosen since the time spend on feature extraction is low and the model accuracy is relatively high. In Table 4.4 timing results can be seen. The model inference is less than 1 second (0.047 s) on RPi and 0.013s on PC. The time taken for the whole feature extraction and inference is 0.768s on RPi and 0.066 s.

4.2.2 2D CNN Model

Table 4.5: 2D Model

Augmentation	Input Size	RAVDESS	EMODB	IMEOCAP
Noise, Shift, Time Stretch	129x129	66.70%	73.60%	62.30%

Table 4.6: 2D Model Speed Test

Model Inference Speed	Only inference	TPU without FE	TPU with FE
Raspberry Pi	0.313	0.271 s	0.325 s
PC	0.031		

2D CNN model is constructed from ourselves. And tested for this thesis purpose. The results show that it obtained 66.7%, 73.6% and 62.3% of accuracy for datasets respectively (Table 4.6). The inference has taken 0.313 seconds on RPi and 0.031s on 0.031s on PC. Overall it took 0.325s to extract Mel-Spectrogram and prediction. The prediction speed is relatively slow because the model is more complex. However, it compensates its model complexity with single feature extraction.

4.2.3 1D CNN LSTM Model

Table 4.7: 1D CNN LSTM Model

Augmentation	Input Size	RAVDESS	EMODB	IMEOCAP
No Augmentation	4 seconds (64000)	62%	61%	52%

Table 4.8: 1D CNN LSTM Model speed test

Model Inference Speed	Only inference	Coral Accelerator TPU
Raspberry Pi	0.642	0.59 s
PC	0.091	

1D LSTM Model is tested in terms of accuracy and speed. The results can be seen in Tables 4.7 and 4.8. It obtained 62%, 61% and 52% of accuracy on datasets respectively. It inferences two times longer than the 2D CNN model (0.642 s) on Rpi and respectively on PC (0.091). Results show that training the LSTM model on raw audio is poorly performing. So, feature extraction is playing a crucial role in performance. Also, the LSTM model appears to be slow on inference. The possible reason for such slow speed is the LSTM architecture by itself.

4.2.4 Self-Attention Model

Table 4.9: Self-Attention Model

Augmentation	Input Size	RAVDESS	EMODB	IMEOCAP
Noise, Shift, Time Stretch	[88]	69.50%	72.60%	70.60%

Table 4.10: Self-Attention Model speed test

Model Inference Speed	Only inference	With FE	TPU with FE
Raspberry Pi	0.501	0.43 s	0.521 s
PC	0.084		

Self-Attention Model is tested in terms of accuracy and speed. The results can be seen in Tables 4.9 and 4.10. It obtained 69.5%, 72.6% and 70.6% of accuracy on datasets respectively. The inference is quite faster than the LSTM Model (0.501 s on

Rpi and 0.084s on PC). The speed with inference and feature extraction is 0.71s on RPi. The results are quite promising. The model is able to show that its performance is better than the LSTM model. However, it uses an additional eGeMAPS extraction step which makes the model slower.

4.3 Modified 1D CNN Model

Table 4.11: Modified 1D CNN Model

Features	Size	Combined Dataset
Only MFCC	[40]	82.3 %

Table 4.12: Modified 1D CNN Model Speed Test

Model Inference Speed	Only inference	With FE
Raspberry Pi	0.04	0.261 s
PC	0.01	0.046 s

In this research a new model is proposed based on the analysis made above. The model architecture is based on the 1D CNN architecture. After the analysis of existing models, it can be seen that the 1D CNN models show the best results in terms of accuracy. So, it was chosen to introduce a slight modification for feature extraction and training process with datasets for this model.

By using only MFCC as a feature and combining all datasets we could achieve 82.3% of accuracy on the combined dataset. And in terms of speed, we could achieve 0.05s of inference and 0.261s of overall time with feature extraction on Rpi.

Table 4.13: Modified 1D CNN Model Speed Test

Model Inference Speed	Only inference	With FE
Raspberry Pi	0.04	0.261 s
PC	0.01	0.046 s

4.3.1 Scenario Simulation and Results

As it was mentioned before, 3 scenarios are proposed to test the RSER Model in terms of time, energy consumption, and accuracy. We have a testbed that contains 196 audio samples from datasets merged to create a stream of audio. Overall, the testbed duration contains 800 seconds of audio. The results can be seen in Table 4.11.

Table 4.14: Scenario analysis

	Prediction Time	Idle state EC	Prediction EC	EC at Sending data
Scinario 1	30 s	2.57 W	0.45 W	0.1 W
Scinario 2		2.57 W		0.2 W
Scinario 3		2.4 W		0.2 W

In Scenario 1, the total time spend for SER is 18 seconds on the edge. At idle state it spends 2.257 W. The energy spent for model prediction is 0.47W. And the energy spent for sending results to sink is 0.1 W.

In Scenario 2, there is no SER inference on the edge. At idle state it spends 2.257 W. There is no energy spent for prediction. And the energy spent for sending audio chunks to sink is 0.2 W.

In Scenario 3, there is no SER inference and VAD on the edge. At idle state it spends 2.4 W. There is no energy spent for prediction. And the energy spent for sending audio chunks to sink is 0.2 W. It is important to mention that the audio chunks are sent continuously (Figure 4.1).

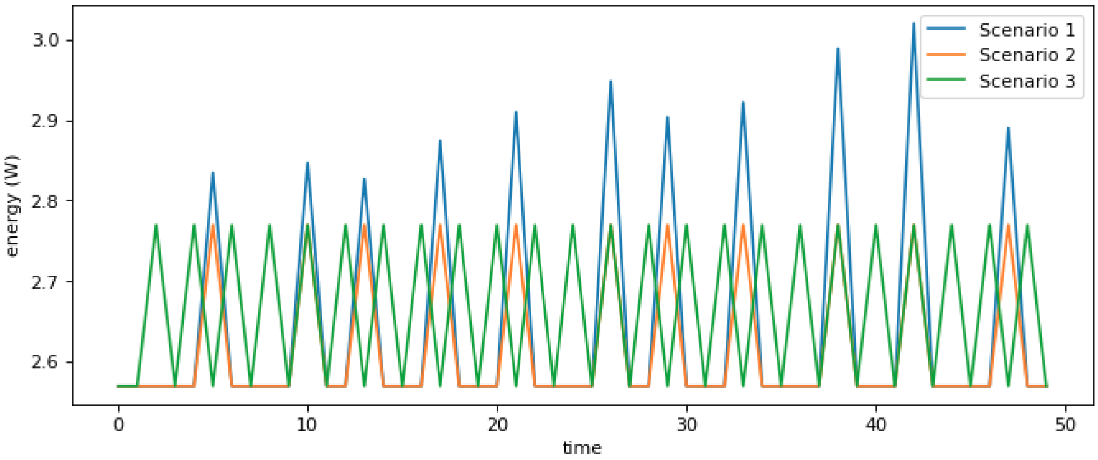


Figure 4-1: Energy consumption at each Scenario

The proposed modified model accuracy achieved 75.5% on a given testbed. The confusion matrix of the proposed model can be seen in Figure 4-2. The model performing significantly well on predicting 'happy', 'angry', and 'fearful' emotion classes. And weak results are shown by 'neutral' and 'surprised' classes. The 'surprised' class is mostly confused with 'happy' class. The model performs well on distinguishing between negative and positive emotions.

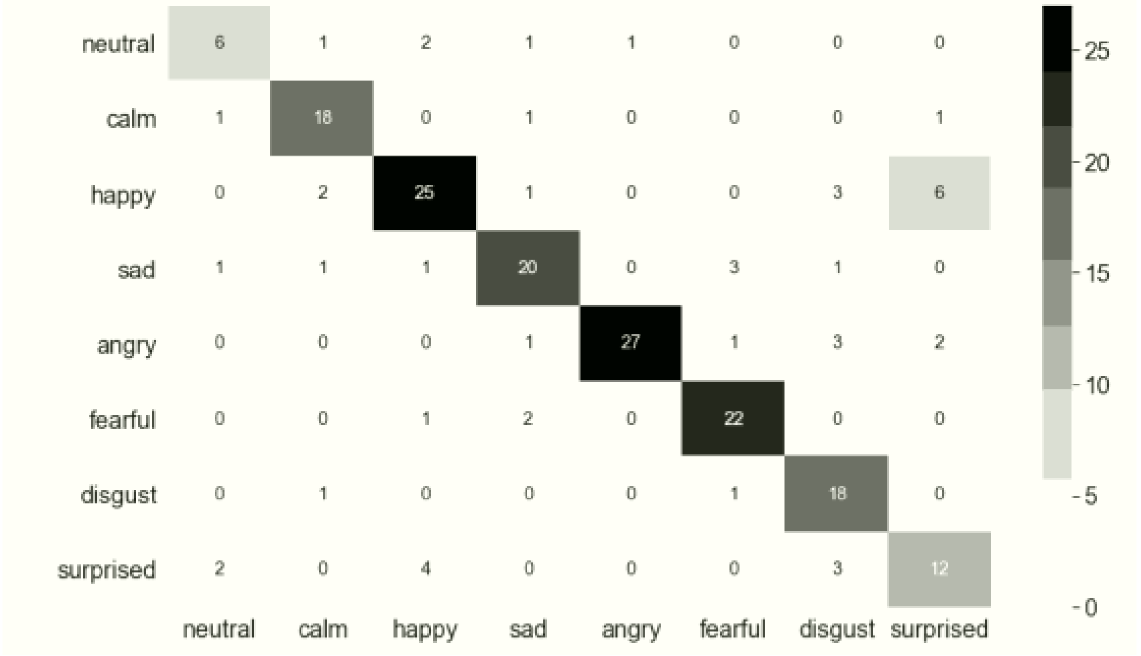


Figure 4-2: Performance of model a on testbed

Chapter 5

Conclusion

This study could introduce an important analysis of the SER models in WMSN. Firstly, we were able to understand that feature extraction methods play a crucial role in model performance. Secondly, simple CNN models with feature extraction methods are predicting more accurately than the LSTM and Self-Attention models. Thirdly, the amount of data have to be large to train simple models.

We could see that processing on the edge consumes more energy than the SER on the sink. However, in a big picture processing on the edge can significantly lighten sink computation burden.

Lastly, we could achieve near real-time speed and good performance on the edge.

Bibliography

[1] W. Lim, D. Jang, T. Lee, Speech emotion recognition using convolutional and recurrent neural networks, in: Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2016 Asia-Pacific, IEEE, 2016, pp.1–4.

[2] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M.A. Nicolaou, B. Schuller, S.Zafeiriou, Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network, in: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2016, pp.5200–5204.

[3] S.R. Livingstone, F.A. Russo, The ryerson audio-visual database of emotional speech and song (ravdess): a dynamic, multimodal set of facial and vocal expressions in North American English, PLOS ONE 13 (2018) e0196391.

[4] F. Burkhardt, A. Paeschke, M. Rolfes, W.F. Sendlmeier, B. Weiss, A database of german emotional speech, Ninth European Conference on Speech Communication and Technology (2005).

[5] J. Zhao, X. Mao, L. Chen, Speech emotion recognition using deep 1d 2d cnn lstm networks, Biomed. Signal Process. Control 47 (2019) 312–323.

[6] Y. Niu, D. Zou, Y. Niu, Z. He, H. Tan, Improvement on speech emotion recognition based on deep convolutional neural networks, Proceedings of the2018 International Conference on Computing and Artificial Intelligence(2018) 13–18.

[7] L. Tarantino, P.N. Garner, A. Lazaridis, Self-attention for speech emotion recognition, Proc. Interspeech 2019 (2019) 2578–2582.

[8] F. Eyben, K.R. Scherer, B.W. Schuller, J. Sundberg, E. André, C. Busso, L.Y.Devillers, J. Epps, P. Laukka, S.S. Narayanan, et al., The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing,IEEE Trans. Affect. Comput. 7 (2015) 190–202.

[9] A. Triantafyllopoulos, G. Keren, J. Wagner, I. Steiner, B. Schuller, Towards robust speech emotion recognition using deep residual networks for speech enhancement, Proc. Interspeech 2019 (2019) 1691–1695.

[10] B.W. Schuller, S. Steidl, A. Batliner, J. Hirschberg, J.K. Burgoon, A. Baird, A.C.Elkins, Y. Zhang, E. Coutinho, K. Evanini, The interspeech 2016 computational paralinguistics challenge: deception, sincerity native language, Interspeech2016 (2016) 2001–2005.

[11] S. Demircan, H. Kahramanli, Application of fuzzy c-means clustering algorithm to spectral features for emotion classification from speech, Neural Comput. Appl. 29 (2018) 59–66.

[12] B. Zhang, E.M. Provost, G. Essi, Cross-corpus acoustic emotion recognition from singing and speaking: a multi-task learning approach, in: 2016 IEEEInternational Conference on Acoustics, Speech and Signal Processing (ICASSP),IEEE, 2016, pp. 5805–5809.

[13] Y. Zeng, H. Mao, D. Peng, Z. Yi, Spectrogram based multi-task audio classification, Multimed. Tools Appl. (2017) 1–18.

[14] D. Issa, M. Fatih Demirci, A. Yazici, Speech emotion recognition with deep convolutional neural networks, Biomedical Signal Processing and Control, 59, May 2020, 101894.

[15] Silva, B.D.; Happi, A.W.; Braeken, A.; Touhafi, A. Evaluation of Classical Machine Learning Techniques towards Urban Sound Recognitionon Embedded Systems. Appl. Sci. 2019, 9, 3885.

[16] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier, and B. Weiss, “A database of German emotional speech,” in Proc. 9th Eur. Conf. Speech Commun. Technol., 2005, pp. 1–4.

[17] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, “IEMOCAP: Interactive emo- tional dyadic motion capture database,” Lang. Resour. Eval., vol. 42, no. 4, pp. 335–359, Dec. 2008, doi: 10.1007/s10579-008-9076-6.

[18] M. H. Moattar and M. M. Homayounpour, "A simple but efficient real-time Voice Activity Detection algorithm," 2009 17th European Signal Processing Conference, 2009, pp. 2549-2553.