Robustness to Noise for Speech Emotion Classification using CNNs and Attention Mechanisms

Lahiru Wijayasingha[a], John A. Stankovic[a]

[a] *Computer Science Department, University of Virginia, United States*

Abstract

Speech Emotion Recognition (SER) is an important task since emotion is a primary dimension in human communication and health. It has a wide variety of practical applications such as assessing the mood of callers to an emergency call center and as a diagnostic tool for therapists. Since most of the SER models in the literature are trained with clean noiseless data and non noise-robust input features, they are not very useful in real world conditions where noise is almost always present. Although there are methods to reduce these adverse effects of noise, a systematic analysis of these methods in the context of SER is lacking. In this paper several different methods to mitigate adverse effects of noise on CNN based SER are developed and analyzed. The SER models trained on the Berlin Database of Emotional Speech were tested with clean data and data mixed with 10 different noise types at different noise levels with signal to noise ratios of 10,15,20,25,30 and 35. We show that the noise robustness of SER models can be improved by combining the magnitude spectrogram with the modified group delay spectrogram, by including synthetic noise in the training data, and by using an attention mechanism. When trained with noisy data, the models trained with the combined input saw a 10% increase in average accuracy than the models using individual inputs and not trained with noise. Adding the attention mechanism to the previous model further improved the accuracy by 5%. Finally, by training and evaluating on the RAVDESS dataset, we demonstrated that the noise robust methods developed can be generalized into other datasets and emotions. We achieved an average accuracy of 81% on RAVDESS dataset under noisy conditions.

Keywords: Spoken emotion recognition, CNN, noise robustness

1. Introduction

Emotion is a mental activity and can be expressed through voice, facial expressions, etc. These emotion expressions form an important part in human communication and health. According to [1] emotions are a primary motivational factor for humans. Emotions can affect a person's perceptions, actions, brain, circulatory systems, health, etc. For example strong anger or fear can raise heart rate by 40 to 60 beats per minute. Therefore, emotions are a contributing factor for medical and mental health problems. Also, research suggests that if emotions like anger, fear and guilt are not attenuated in time, they could cause various mental health problems. Fear or anxiety is accompanied by changes in corticosteroid levels in the blood which is related to numerous physiological processes [1] such as stress response and immune response.

Automatic emotion recognition (AER) can be used to understand emotions of people [2] and is necessary for interaction between human and machines such as robot health assistants [3]. There are many applications for emotion understanding in health such as diagnostic tools for therapists, helping caregivers of dementia patients, helping post traumatic stress disorder patients, assessing the emotional state of callers to a emergency call center, and even outside of health such as to assess emotional state of drivers in cars [3] and for media [4].

AER systems can be built around different modalities such as visual and auditory. Visual AER systems use cameras and considers the visual aspects of human emotions (for example facial features). Auditory systems on the other hand use microphones and try to determine the users emotional state from their voice. In this research we focus on auditory systems which can be generally called Speech Emotion recognition (SER) systems. They can be applied where cameras are not available and/or not appropriate (for example due to privacy reasons). For instance, when assessing the emotional state of a caller to an emergency call center the visual modality is usually not available.

One of the weaknesses of the state-of-the-art SER systems is they are very sensitive to noise. These noises are caused by other sound producing sources in the vicinity of the SER system which can corrupt the speech signal the SER system is analysing. But only very limited amount of research was performed addressing this problem in the context of SER. Therefore, in this paper, we develop, analyze, compare and propose several potential solutions to solve this problem.

Convolutional Neural Network (CNN) models were used for SER since they have the state-of-the-art performance in the literature. Several spectrogram based feature types as inputs to the CNN models were used, compared and combined in this study. All of these spectrograms were obtained by performing Discrete Fourier Transformation (DFT) on the speech signals. Traditionally, SER models use only magnitude spectrograms for audio classification. But in this study we also use Modified Group Delay (MGD) spectrograms [5] and unwrapped phase spectrograms obtained with python package scipy [6].

The main contributions of this paper are:

1. Majority of state-of-the-art SER models are CNNs and use magnitude spectrogram as the input. But performance of these models degrade significantly with background noise. There are evidence from literature that Modified Group Delay (MGD) may be robust under noise. We combine magnitude spectrogram with MGD spectrogram and show that the Fully Convolutional Neural Network (FCNN) models trained with this combined input is more robust to noise than just using magnitude spectrogram. We perform initial experiments on the Berlin Database of Emotional speech. Using other techniques such as training with artificially added noise and attention mechanism alongside the combined input we show that an average improvement of 15% accuracy (F1 of 0.16) can be obtained under noisy conditions when compared to a traditional model which only uses magnitude spectrogram as input.
2. We show that including synthetic noise in the training data improves the noise robustness of all the models considered. Interestingly, the model with combined magnitude and MGD spectrograms as inputs saw a larger improvement than those used individual spectrogram inputs. Doing this step improved the accuracy by 10% (F1 by 0.11) over the model which used magnitude spectrogram alone and did not train with noisy data.
3. Including an attention mechanism to the FCNN model with combined input and training with noisy data also improved the noise robustness by an accuracy of 5% (F1 by 0.05). We also provide evidence that attention mechanisms can ignore noisy data sections of speech and pay more attention to important and cleaner sections of the speech.
4. Our best model showed an average accuracy of 76% (F1 of 0.73) over all the noise levels (Signal to noise ratios from 10 to 35 and clean speech) and all the noise types considered. If the performance under AWGN is considered our model had an accuracy of 76% over all the noise levels (including clean speech). This performance is a significant improvement over the model mentioned in [7] which reports an accuracy of 56%.
5. The above results were obtained with the Berlin Database of Emotional speech. Next the best fine-tuned model architecture, noise robust features and the hyperparameters chosen from those steps were used and trained on the RAVDESS dataset [8]. This model obtained an accuracy of 91% (F1 - 0.91) under clean speech and 81% average accuracy (F1 - 0.81) under all noise types and levels considered. This shows that our solution can be generalized to other datasets and emotions.

The remainder of this paper is organized as follows. First we discuss some related work and preliminaries in section 2. Next, we discuss the methods and solution framework that was used in Section 3. Then, we present the experiments and results in Section 4. A discussion and conclusions are presented in Sections 5 and 6.

2. Related work and preliminaries

2.1. *CNNs for SER*

Deep learning has become the state-of-the-art of many audio classification tasks including SER [2]. Within the deep learning domain, CNNs have become popular for computer vision tasks [4]. But if any set of features can be represented as images or stack of images (multi dimensional arrays), CNNs can be used to classify them. Magnitude spectrograms obtained via Discreet Fourier Transformation (DFT) of audio signals is commonly used as the input to CNNs in audio classification solutions. Because spectrograms are 2D images, the audio classification problem may be treated similar to a regular image classification problem using a CNN [4] [9] [3] [10]. [10] showed that CNNs are better suited than Deep Neural Networks (DNN) and Long Short-Term Memory (LSTM) networks for the SER problem. Although it is not straightforward to compare state-of-the-art models in SER due to the variety of datasets and evaluation methods, CNNs seems to be performing better than other types of classifiers for SER.

2.2. *Inputs for CNNs*

In terms of CNNs various types of inputs qualify for the task of SER. [11] found evidence that frequency domain features are better suited than time domain features for acoustic event classification using deep learning. Various frequency domain inputs such as DFT components [11], magnitude spectrograms [12] [3] [13] [9] [14] [10] [15] [16] [2] [17] and wavelet features [18] have been used in the literature. Spectrograms can retain more information than most hand-crafted features and are low dimensional than raw audio [9] and are one of the most commonly used feature types for audio related tasks.

Spectrograms are obtained by performing Discrete Fourier Transformation (DFT) on the audio signal. DFT of a signal around the sample n can be represented by

$$X_n(\omega) = \sum_{m=-\infty}^{+\infty} x(n)w(n-m)e^{-j\omega m} \tag{1}$$

Here the frequency of the component $X_n(\omega)$ is ω and w is a windowing function. $X_n(\omega)$ is a complex number and can be represented in terms of magnitude and phase angle as.

$$X_n(\omega) = |X_n(\omega)|e^{jarg[X_n(\omega)]} \tag{2}$$

Here $arg[X_n(\omega)]$ is the angle between imaginary and real parts.

If DFT of a signal was taken with sliding windows a distribution of frequency characteristics along time axis can be obtained. This is called a spectrogram. If the magnitude part $|X_n(\omega)|$ from equation 2 is used, this is called the magnitude spectrogram. If the part $arg[X_n(\omega)$ is used, it is called a phase spectrogram. To recreate the original signal from the DFT components, both magnitude and phase parts of the signal should be used [19].

But traditionally only the magnitude spectrogram is used and the phase spectrogram is ignored [20]. This is mainly due to the fact that the phase is discontinuous due to mathematical problems occurred when calculating the inverse of trigonometric functions. A simple correction called phase unwrapping should be performed to mitigate this problem. Researchers are beginning to explore the effectiveness of the phase based features for various tasks such as speech and speaker recognition. They found that combining magnitude with phase spectrogram improves the performance [21] of SER systems. [22] deals with detecting gunshots from audio data. They calculate statistical features from magnitude and phase spectrogram of the audio signal and conclude that combining these features improves performance than just including features from the magnitude spectrogram. [23] lists different phase representations those are being used in literature. They are relative phase shift, time-frequency derivatives of phase, phase dispersion and phase distortion.

Modified Group Delay (MGD) is another quantity that has seen a recent rise in usage. It combines magnitude and phase components of the signal. There exists some research that investigate the effectiveness of MGD for speech and speaker recognition. But only a limited number of literature investigate this for emotion recognition. Research shows that MGD performs better than traditional features for whispered emotion recognition [5]. Modified Group delay can be calculated using the DFT. Let

$$y(n) = nx(n) \tag{3}$$

Here $x(n)$ is the n^{th} sample of the speech signal. If $X(\omega)$ and $Y(\omega)$ are the Fourier transformations of $x(n)$ and $y(n)$ at any given n
Define

$$\tau'(\omega) = \frac{X_R(\omega)Y_R(\omega) + X_I(\omega)Y_I(\omega)}{|S_c(\omega)|^{2\gamma}} \tag{4}$$

Here X_R and Y_R are real parts of X and Y. X_I and Y_I are imaginary parts of X and Y. To obtain $S_c(\omega)$, first squared magnitude $|X(\omega)|^2$ of the signal $x(n)$ is obtained and then it is cepstrally smoothed. Modified Group Delay (MGD) can be obtained as follows.

$$\tau(\omega) = \frac{\tau'(\omega)}{|\tau'(\omega)|}(|\tau'(\omega)|^{\alpha}) \tag{5}$$

In these equations α and γ are parameters where $(0 < \alpha \leq 1)$ and $(0 < \gamma \leq 1)$. These parameters should be tuned depending on the application. Obtaining MGD with a moving window yields a MGD spectrogram.

Although many frequency domain feature types are being used in literature, their comparative performance and the reason to choose a particular feature type is rarely studied [24]. But there is evidence that different features can yield different performance levels. Even slightly changing the representation of features can change the performance. For example, [17] use CNNs for snore sound classification. They input magnitude spectrograms obtained with 3 different color maps. Different color maps gave different performance.

Previous studies have considered combining different feature types as inputs to deep learning based audio classification models. [14] [21] and [18] found that combining various frequency domain features improved the performance of their audio classification models. There are different methods to combine feature types as inputs to CNN models. [14], [25] and [26] combines each different spectrogram as a different channel when they are input to the CNN. This is similar to the way R,G and B channels are treated in RGB images. Different from this method, [21] and [26] concatenates two different spectrogram representations side-by-side to form a single image. Although it is intuitive to assume that different methods of combining the features may yield different results very limited research is done in this regard.

2.3. CNNs accepting variable input sizes

The height of a spectrogram depends on the range of frequency considered and the length depends on the time interval considered. Since people may speak utterances of variable length, an emotion classification models should have the capability to handle inputs with variable lengths. But traditional CNNs only accept a fixed size of input.

One method to overcome this is to combine a CNN with a Recurrent Neural Network (RNN) model [2] [4] [13]. In these studies the CNN acts as a feature extractor and the RNN (in this case it is a LSTM) layer learns the time sequence relationships in these features. [2] showed that combining CNN with LSTM better performance can be obtained than just using a CNN. This may be due to the fact that LSTM can model the sequential characteristics of the input.

Another method to handle variable input lengths is to use a Fully Convolutional Neural Network (FCNN). A FCNN does not explicitly model the sequential nature of the input. Instead it considers the whole input (of variable length) and make inferences. A FCNN can be generated starting from a traditional CNN by replacing all fully connected layers with convolution layers. This can also be beneficial because it reduces the number of trainable parameters in the CNN. [27] uses FCNN for spoken emotion recognition task and saw that it outperformed a model which combined CNN and LSTM.

2.4. Performance under noise

SER systems can be adversely affected by environmental distortions. Noise and reverberation are two main sources of distortions that could degrade the quality of the audio signal. Reverberation depends on the characteristics of the environment that the sound is produced. Noise is generated due to sound sources other than the one we are interested in. When the distance form the speaker to microphone increases, increased noise relative to signal can be observed [28]. Noise and reverberation can affect the original clean speech signal as shown in equation 6.

$$y(t) = h(t) * s(t) + n(t) \tag{6}$$

Here h(t) is the room impulse response between the microphone and the speaker. s(t) is the clean speech and n(t) is the background noise. * is convolution operation. Note that reverberation is represented as convolution with the original signal. Noise is represented as addition [29].

Studying both reverberation and noise for SER is important because both of these distortions can affect SER systems in an adverse manner. Previous studies have attempted to study/reduce these effects from both noise and reverberation. For example [30] studied the effects of reverberation and [28] studies the effects of both noise and reverberation for SER. [16] use CNN with amplitude spectrograms as input and observed performance degradation when noise is added to the inputs. [12] use magnitude spectrogram and provides reasoning for spectrograms may be better for handling noisy data. [31] studies the effects of the distance to the microphone from the speaker on a SER system because increasing the distance also increases the amount of noise with respect to that of the signal.

In this study we aim to find solutions for the adverse effects of noise on SER systems. Although the problem of reverberation is also very important, certain solutions for that problem can be obtained easier than the problem of noise. For instance, if we have prior knowledge that the SER system will operate in a certain environment (e.g. living room of a house) it can be assumed that majority of the living rooms have similar reverberation characteristics. Therefore if the SER system was trained with data recorded in living rooms, this SER may be able to solve the problem of reverberation for operation inside similar living rooms. The problem of background noise on the other hand is harder to solve. Although there are certain types of noises such as air conditioner which is present in most of the house holds, there are many other unpredictable noises such as a noise of a vehicle outside the house or a ring of an alarm. Therefore it is not realistic to assume that all the possible noise conditions can be known beforehand. Therefore we focus on solving the problem of background noise in this study.

2.5. Methods to mitigate effects of noise

Several methods can be used to reduce the adverse effects of noise on SER systems. Some of these methods are described below.

One method is to remove the noise from raw speech signal or calculated features. This is generally called speech enhancement. Several main categories of speech enhancement can be identified from literature. They are spectral subtraction based [13], subspace based, statistical-model-based, Wiener-filter based algorithms [32] and de-noising autoencoders [33] [34]. The aim of this research is to find noise robust features, CNN architectures and methods of training them. Once these are found, speech enhancement can also be applied on top of our solution to further improve the performance. Future studies may be necessary to quantify these further improvements.

The second method is using features which are robust to noise. Some features are more robust to reverberate and noisy conditions than others [28]. According to [35] different frequency domain features may have varying robustness for noisy data for speech recognition task. The noisy environments they consider are background music, white noise and reverberate environments. [35] use 1st and 2nd derivative of Mel-Frequency Cepstral Coefficients (MFCC) features for speech recognition. They found that including the derivatives of MFCC features improves performance in both noisy and quiet environments. [19] described MGD which is calculated using both magnitude and phase of the Fourier transformation of the signal. MGD is analytically shown to be robust under additive noise [36] [19] [37]. Practically, features calculated with MGD is shown to be robust under noise for speech recognition [37] [19] and for voice activity detection [36]. We explore this dimension in this study. Only a very limited number of research is done regarding noise robust features in the context of SER. We explore the noise robustness of several DFT based features and their combinations.

The lack of performance of models under noisy conditions can be attributed to the difference between training and testing data distributions. Researchers have tried to reduce this problem by trying to bring the training data distribution closer to that of the testing data by artificially injecting noise or reverberations to the training dataset. [28] and [30] utilize this method to improve model performance under reverberate conditions. The same methods can be applied to noisy conditions. Limited number of research is done exploring this dimension with regard to SER systems. In this study we explore the effect of adding synthetic noise to training data on the noise robustness of these models.

There are evidence that certain NN architectures are more robust to noise than others. For example [38] shows that CNN with attention mechanisms could be more robust to noise compared to other types of models (e.g. LSTM). Attention mechanism used with CNNs can be used to improve the model performance by teaching a model to explicitly pay attention to important parts of the input and to ignore unimportant parts. [39] used attention mechanism with a CNN for image captioning tasks and observed an improvement in performance. Similar models can be used for SER [27]. [27] uses FCNN with attention mechanism and saw an improvement of performance compared to other models. But a detailed analysis on the noise robustness of CNNs with attention mechanism was not performed in the literature. In this research we implement an attention mechanism for a Fully Convolutional Neural Network (FCNN) and show that adding the attention mechanism improves the noise robustness properties of the model.

3. Methods and solutions

The aim of this study is to find features, training procedures and model architectures that are robust to noise for the task of SER. This section describes the methods used to achieve this, problems faced and solutions.

3.1. Data Sets

For the first sections of this study audio speech from the Berlin Database of Emotional Speech is used [40]. It contains around 500 utterances spoken in German. The emotions happiness, anger, anxiety/fear, disgust, boredom, sadness and neutral are present in the dataset. First the hyperparameters of the models were tuned and appropriate input features were selected using the Berlin dataset. Afterwards the model with those hyperparameters and input features were tested on the speech audio data from the RAVEDESS dataset [8]. This dataset contains two intensities of emotion expression; strong and normal. Only the strong emotional expressions were used. For this study, the emotions calm, happy, sad, angry, fearful and surprised were used from the RAVDESS dataset. 575 utterances were present in the selected data from RAVDESS.

3.2. Noise

Noise types used. Different types of noise can affect the performance of a SER system. But an analysis of SER performance under different noise types is not performed in an adequate manner in the literature. Therefore, a set of 9 common indoor noises were selected for this study from the Freesound Dataset [41]. They are the sounds of crickets, alarms, kettle whistling, rain, steps (person walking), thunder with rain, traffic noise, vacuum cleaner, and air conditioner. In addition, additive white Gaussian noise (AWGN) was used.

These noise types are selected because they are very common sources of noises in indoor environments. For example, air conditioner may be always on during summer. Also sounds with various characteristics were selected. For example, unlike air conditioner hum, steps is of intermittent nature and alarm sound has a higher frequency.

Additive white Gaussian Noise. This kind of noise can be added (arithmetic element-wise addition) to the signal. Also its mean value is zero (randomly sampled from a Gaussian distribution with mean value of zero; standard deviation can vary). It contains all the frequency components in an equal manner. AWGN is easier to model and easier to generate. Since AWGN contains noise equally in all the frequency bands, researchers (e.g. [42]) use it to approximate different types of noises and to compare the performance of different models and features.

The SER models trained during this study were evaluated with all of these noise types. It is hypothesized that a model which is competent under these noise conditions will be able to handle a wide variety of indoor noise types which are not used here.

Signal to noise ratio. For the experiments in this study, one of the qualities that was evaluated is the performance of a classifier under noisy conditions. Signal to noise ratio (SNR) is a method to quantify how much noise it present in a signal. For the purposes of this study, signal is the speech. Noise is one of the noise sources described in the last section. SNR can be defined as follows.

$$SNR = 10log(\frac{RMS^2_{signal}}{RMS^2_{noise}}) \tag{7}$$

where RMS_{signal} is the RMS value of signal and RMS_{noise} is that of noise. log represents the logarithm of 10. In the experiments performed, noise was added to the clean signal at different SNR levels and the performance of models were observed.

3.3. Feature selection

Magnitude spectrogram is the most commonly used input type for CNN based SER systems. So, it will be included in this study. Since there are evidence suggesting including phase information makes models perform better, we also use unwrapped phase spectrogram as an input. Also, since modified group delay (MGD) was theoretically proven to be robust to noise as described in section 2.5 we include MGD spectrogram in our study. All 3 of these input types are Fourier transform based.

Different input types and their combinations may have different noise robustness characteristics. Therefore, models were trained taking each of these input types and some of their combinations as inputs and their performance under noise is compared. This way, the best performing input types/input combinations can be selected.

3.4. *Creating the FCNN architecture*

To find the best CNN architecture for each feature type, the auto ML package Autokeras [43] was used. To build the classifier with Autokeras, ImageClassifier class with $max\_trials = 20$ and fit function with $epochs = 20$ were used. An interesting observation is that the optimum CNN architecture found by Autokeras was the same for all the situations (Different input types and their combinations). After obtaining a CNN architecture, it was converted into a FCNN by replacing all the fully connected layers with convolutional layers. The architecture is shown in Table 1. Note that there are 7 filters in the last convolutional layer. These filters correspond to the 7 different emotion classes in the Berlin dataset which is used in initial experiments.

Table 1: FCNN architecture

Layers
conv2D 32 3x3 1
Activation relu
Max Pooling 2x2 2
conv2D 64 3x3 1
Activation relu
Max Pooling 2x2 2
conv2D 16 3x3 1
Activation relu
Dropout 0.5
conv2D 7 3x3 1
Global Average Pooling
Activation softmax

3.5. *Calculating spectrograms*

First, start and end silence sections were removed from voice samples. All the voice samples were re-scaled so they will be between $[-1, 1]$. Then features were calculated from these samples. Note that the length of these voice samples varied since FCNN can handle variable length inputs. Note that MGD has the parameters α and γ from Section 2.2. After some trial and error $\alpha = 0.6$ and $\gamma = 0.5$ was selected and kept constant throughout this study.

The input types used are magnitude, MGD and unwrapped phase spectrograms. To calculate these, the voice signal was broken into blocks of desired length (depending on the desired DFT window length) and then applied a hamming window. Afterwards DFT was performed. From the DFT results, magnitude, MGD and unwrapped phase were calculated. DFT was performed with python library scipy.

DFT window length. Different lengths for the blocks (this is called DFT window length in this study) were chosen. The frequency and time resolution of the spectrograms depend on the chosen DFT window length. Larger lengths results in higher frequency resolution and lower time resolution and vice-versa. Therefore different DFT lengths may react to various noises in different ways. The noise performance of the models may depend on the DFT window length chosen. These lengths are chosen to be 25 ms, 50 ms, 75ms and 100ms. A separate model for each of these DFT window lengths were trained.

3.6. *Combining Input spectrograms*

The goal of this section is to find the effects of combining different inputs on noise robustness of SER. It is hypothesized that different methods of combining different inputs may yield varying levels of SER performance and combining inputs may provide better results than just using individual inputs. To test this some input types were combined in two different ways and compared.

Two different methods of combining two feature types were used in this study. They are combining them as different channels of an image (called comb1) and combining them side-by-side (called comb2). Figure 1 shows the two methods of combining feature types. If both $Feature\_1$ and $Feature\_2$ are 2D arrays of dimension $h * w$, the dimension of comb1 would be $h * w * 2$ and dimension of comb2 would be $h * 2w$. A CNN can be trained using regular methods with both of these input types.

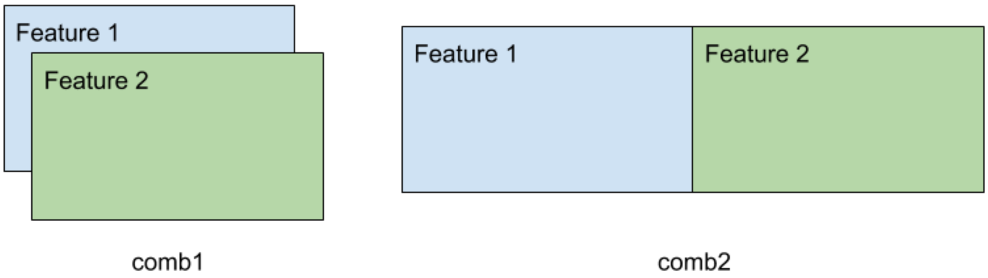

Figure 1: Combining Features

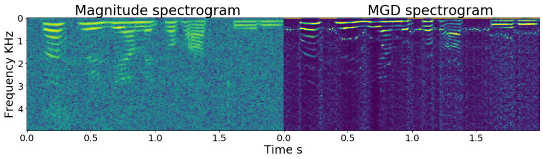

Figure 2: comb2 with AWGN

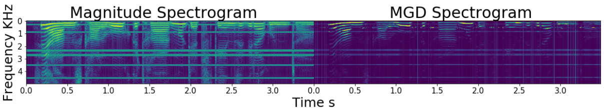

Figure 3: comb2 with noise at several frequency bands

3.7. *Model training*

All the FCNN models were trained with Keras deep learning library with the optimizer adam. The starting learning rate was 0.001. To train all the FCNN models except under batch training in section 4.5 and when training on RAVDESS in section 4.8 batch size of 1 was used because keras can only have fixed length inputs in the same batch. For details on the batch training refer section 4.5. Learning rate was decayed by $1 * 10^{-6}$ once every epoch. Class weights were used while training due to the class imbalance. These class weights were inversely proportional to the number of instances present in each class.

3.8. *Model testing*

In the initial experiments, for each feature and DFT window length 10 fold cross validation was performed on the Berlin Database of Emotional Speech. First each model was evaluated under clean speech test set. In order to evaluate the noise robustness of these models, they were evaluated under clean speech clips mixed with various noise types at various noise levels (measured by SNR). SNRs used are 10, 15, 20, 25, 30 and 35. For example when evaluating a model under noise of crickets, cricket noise was mixed to the original clean test data set at different SNRs. The same procedure was performed for all the different noise types including AWGN. Afterwards the model accuracy and F1 score was obtained. F1 score is used to report most of the results because it is more useful for evaluating datasets with class imbalance. For the RAVDESS dataset, train data was prepared with 80% of the data and testing was performed on the rest. Procedure for testing under noisy conditions was identical to that of the Berlin dataset.

3.9. *Training with noise*

One reason that the models perform worse on the testing data is that the distributions of train and test data are different. If models are trained with clean speech and used in real world noisy environments, the model performance degrades because the test data contains noise which was not present in train data. To bridge this gap, noise may be included in the training data. It is hypothesized that by injecting noise to training data, the performance of the model on test data under noise can be improved.

To prepare the training data, 3 data sets were generated from the clean speech data. One is the clean speech data itself. A second data set was prepared by injecting AWGN of SNR 40 to clean speech. Figure 2 shows one instance of these samples. A third data set was created by injecting random noise at several random frequency bands. Figure 3 shows one of the samples with this type of noise. Note that Figures 2 and 3 show magnitude and MGD spectrograms combined together with the method comb2.

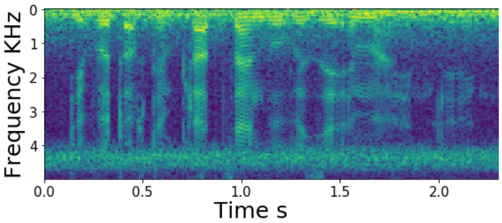

Figure 4: Magnitude spectrogram with cricket noise

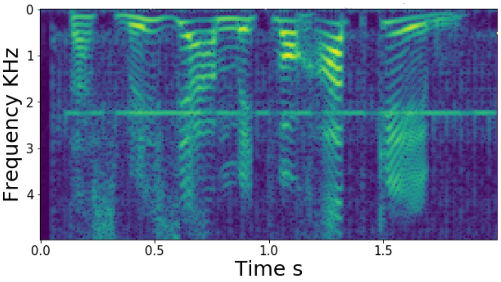

Figure 5: Magnitude spectrogram with alarm noise

3.10. Adding attention mechanism

An attention mechanism was added as shown in the Figure 6. It functions according to equations 8 and 9. This attention mechanism is influenced by and modified from the one in [39]. [39] implements the attention mechanism for a traditional CNN architecture with fully connected layers. This had to be modified to fit FCNN which only contains convolution layers as feature extractors. A convolution layer is used for extracting attention weights. There was no previous research done regarding combining FCNN models with attention mechanism and evaluating its noise robustness. Although [39] used both spatial and channel-wise attention this study only uses spatial attention due to increased complexity involved in implementing both types of attention at the same time. Due to the smaller nature of the datasets used in this study, it can be imagined that increasing the number of model parameters will increase over fitting.

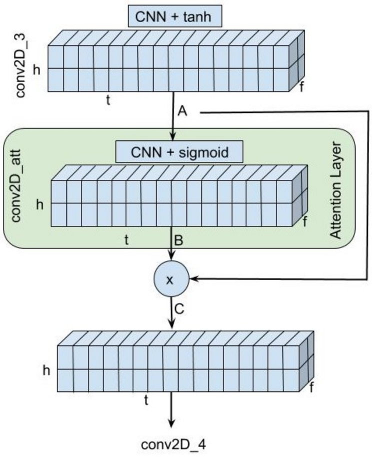

Figure 6: Attention Mechanism

$$B = sigmoid[conv2D_{att}(A)] \tag{8}$$

$$C = A * B \tag{9}$$

Attention mechanism can be used to teach the model in an explicit way to focus on relevant sections of the input and ignore irrelevant parts in the time-frequency space. The motivation behind using attention mechanism to create noise robust SER models is that certain noises affect localized regions in the time-frequency space. For example Figures 4 and 5 show voice clips mixed with cricket and alarm sound. Majority of these noises are contained in some particular frequency ranges. Alarm noise in this example corrupts a narrow frequency range around 2.1KHz. Cricket sound has corrupted some lower frequencies and higher frequencies around 4-5KHz. If the model learns through attention mechanism to focus on clean sections of the input and ignore the noisy regions it is hypothesized that the model will be more robust to these types of noises.

4. Experimental Results

4.1. Training FCNN

The FCNN architecture shown in Table 1 was trained with spectrograms generated from voice recordings of variable lengths. Figures 7, 8 9 and 10 shows the performance of the FCNN models under various levels of noise. The SNRs used are 10,15,20,25,30 and 35 as mentioned in section 3.8. In Figures 7, 8 and 9 at each SNR, the average performance of each model under all of the noise types mentioned in section 2.2 are shown. A separate model was trained for each DFT window length and feature type. The DFT window length used is shown in the legend. Figure 10

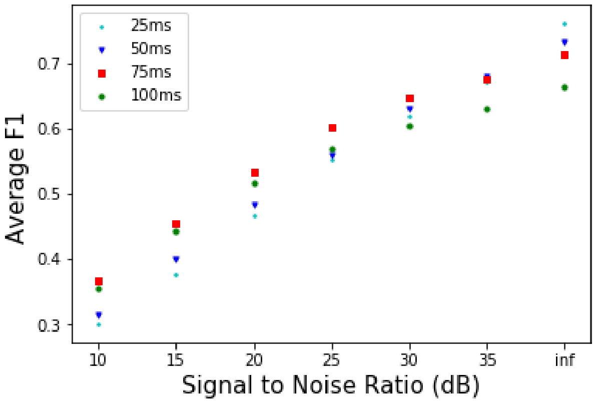

Figure 7: Performance of magnitude models under average noise

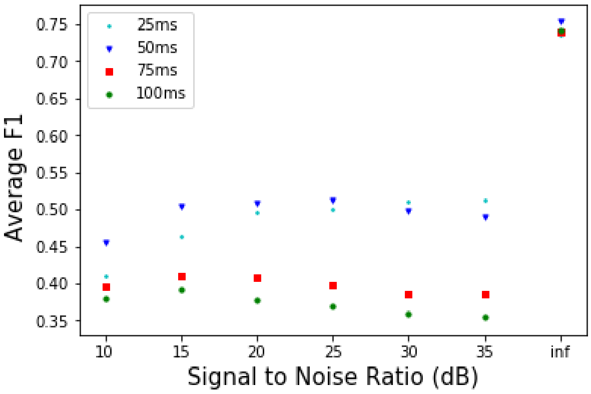

Figure 8: Performance of MGD models under noise

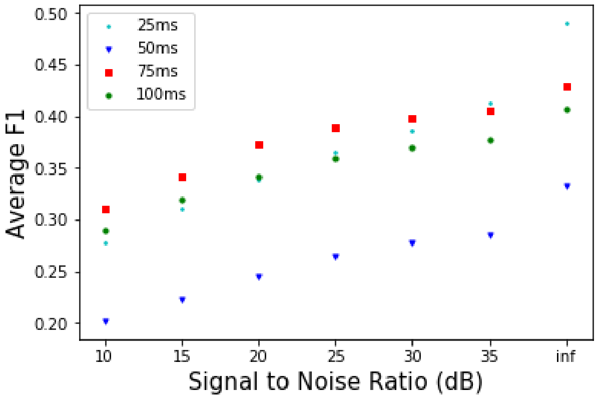

Figure 9: Performance of phase models under average noise

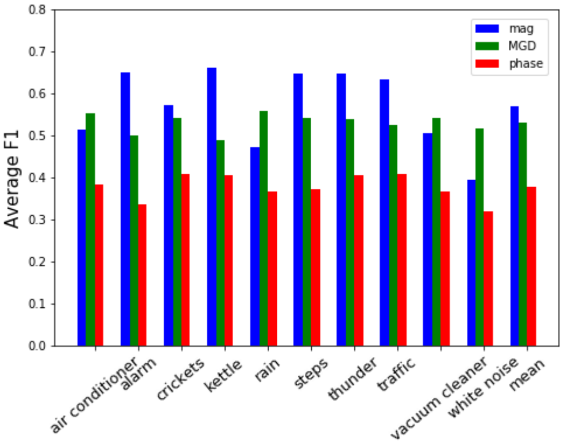

Figure 10: Comparative performance under all types of noises

shows the performance breakdown under different noise types for the best model trained under each feature. For each of these noise types the average performance under the various SNR were taken.

Figure 7 shows the mean accuracy of the FCNN model trained with magnitude spectrogram as input. The model trained with DFT window length of 75ms shows the best overall performance under noisy conditions. Under clean speech, it shows an F1 score of 0.71. Interestingly the worst performing model under noise (25ms model) performs best under clean speech. The performance levels of all the models drop significantly with the addition of noise. Figure 8 shows the performance of the MGD model. 50ms model shows best performance under noise and clean speech. It shows an F1 score of 0.75 under clean speech. Figure 9 shows the performance of the model which uses phase spectrogram as input. The best overall performance was achieved when DFT window length is 75ms and was 0.42 under clean speech. Figure 10 compares the best models from each feature. Here the magnitude, MGD and phase features were derived using DFT window sizes 75ms, 50ms and 75ms respectively.

From these plots it can be seen that different features may have different performance for clean speech and under various noise levels. Also using different DFT window length may yield different performance both for clean and noisy speech. The DFT window length which gives the best performance under clean speech may not be a good choice for SER under noisy speech. From the features evaluated in this section, MGD shows the most significant drop of performance going from clean speech to noisy speech. For example the performance of the 50ms MGD model drops from 0.75 to 0.49 when going from clean speech to speech with SNR of 35. But its performance is very stable around 0.50 until SNR is 15. From Figure 10 it can be seen that for certain noise types MGD performs better and for

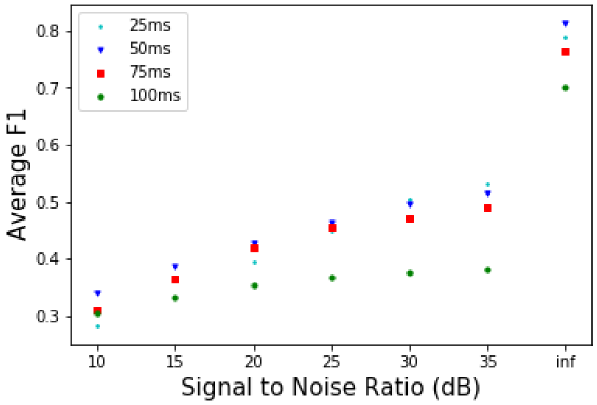

Figure 11: performance of comb1 models under noise

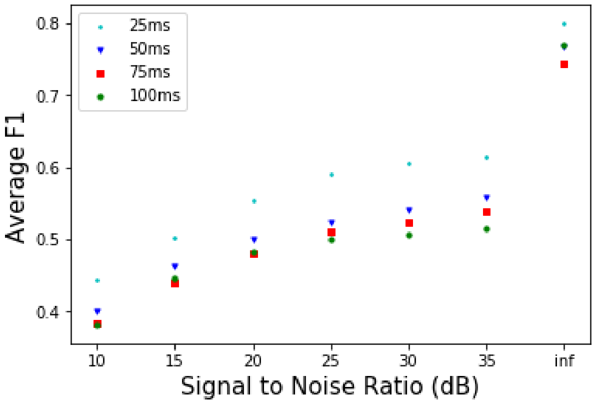

Figure 12: performance of comb2 models under noise

the others mag is better. MGD performs better under noise types air conditioner, rain, vacuum cleaner and white noise. Note that these noise types are more similar to white noise and corrupts broad frequency bands when compared to other types of noises used here. Phase always performs the worst under all the noise types. Therefore only magnitude and MGD features were selected for further analysis. From this section it can be seen that not only magnitude but also phase based features like MGD can be used for SER. MGD models perform better than magnitude spectrogram models under certain types of noises.

4.2. Combining spectrograms

Next the effects of combining different input types to FCNN were studied. These input types were combined according to two methods (comb1 and comb2) as discussed in section 3.6. From the previous section it can be seen that the performance may depend on the selected DFT window length. So for each combination method different DFT window lengths were used to train and validate the models.

From Figure 11 it can be seen that the DFT window length 50ms performs better for combination method comb1. Under clean speech comb1 model with 50ms DFT window length yields a F1 score of 0.81. For comb2, 25ms is the best DFT window length as can be seen from Figure 12. The performance of this model under clean speech is 0.8.

Next the best models for magnitude spectrogram, MGD spectrogram, comb1 and comb2 input types were compared. Figure 13 and Figure 14 shows the relative performance of these models. Figure 13 was obtained by taking the mean performance of the best magnitude, MGD, comb1 and comb2 models over different SNR under all the noise types. Values in Figure 14 were obtained by taking the mean value of performance under all SNRs under individual noise type.

Although the best performing model under clean speech is comb1 which gives a F1 score of 0.81 according to Figure 13, it never performs the best at any noisy condition (see Figure 14). Also comb1 is very sensitive to noise since its F1 drops to 0.51 under a little addition of noise having a SNR of 35. Therefore, we can safely eliminate comb1 from further consideration. According to Figure 13, the mag model shows an F1 score of 0.71 under clean speech. It has the best performance under noisy conditions among all the models until SNR drops to 20. But from Figure 14 it can be seen that comb2 model performs the best when we take the mean performance over all the noise types. Also a different kind of model may be the best one under a different noise type.

From the results of this section, it can be concluded that by combining phase based features such as MGD with magnitude, models more robust to noise can be built (compared to model just using magnitude as input). Also different methods of combining features can have different levels of noise robustness. Under these conditions comb2 method seems significantly better than comb1 method of combining magnitude and MGD features.

4.3. Training with noisy data

Previously, all the models were trained with just clean data. In this section the effects of training with noisy data is studied. Three best performing models from previous section (mag, MGD and comb2) were selected and retrained

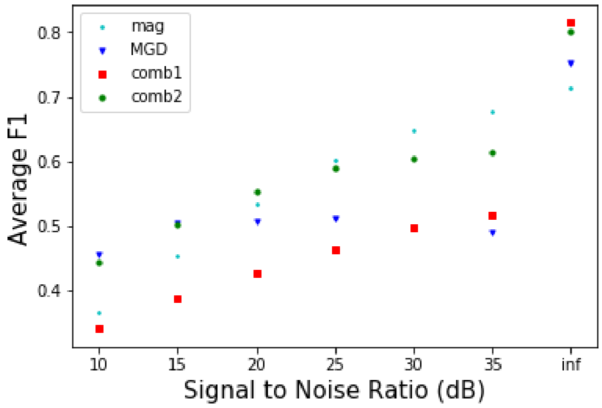

Figure 13: comparative performance of models under noise

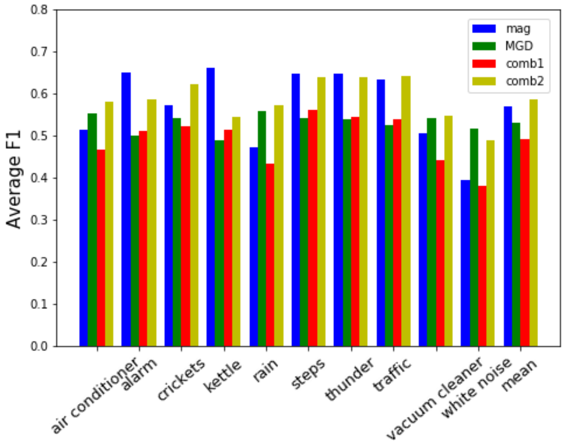

Figure 14: comparative performance of models under noise

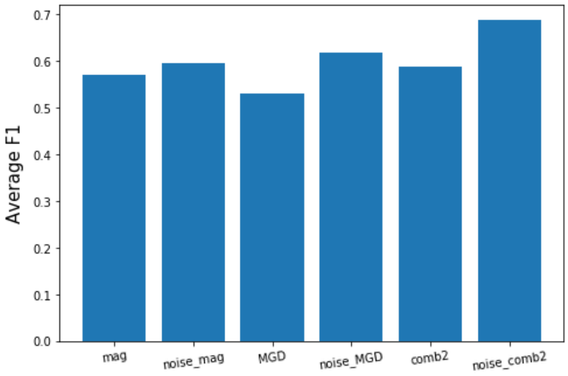

Figure 15: Improvement when training under noise

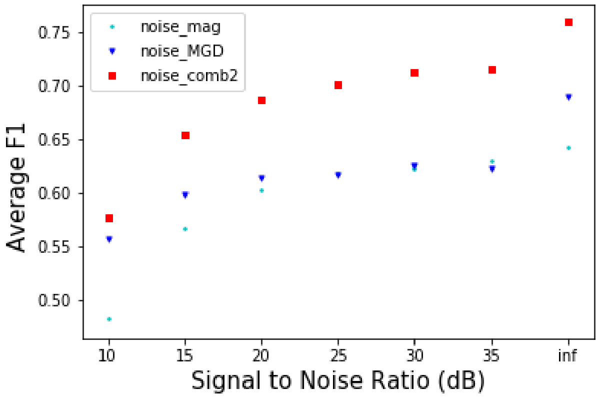

Figure 16: Mean performance when training under noise

with data mixed with artificial noise as mentioned in section 3.9. When trained with noise, mag, MGD and comb2 models were named noise_mag, noise_MGD and noise_comb2. Figure 15 shows the improvement obtained by training with noise for each model. This figure shows the average performance under all noisy conditions and all SNRs. All the models saw an improvement when trained with noise. F1 score of magnitude model improved from 0.57 to 0.59. MGD model improved from 0.53 to 0.61 and comb2 improved from 0.58 to 0.68.

Figure 16 shows the mean performance of models under all the noisy conditions under various SNRs. The model trained with magnitude spectrograms perform the worst in general and noise_comb2 performs the best. Figure 17 shows the performance of the models under all the different noise types. For each noise type, the average over all the SNRs were taken. It can be seen that noise_comb2 performs the best for each and every individual noise type.

From the results of this section, it can be concluded that training with artificial noise, the noise robustness of SER models can be improved. Furthermore, the improvement is greater when the model takes both magnitude and MGD inputs compared to just using one of them. One other interesting observation from Figure 17 is that noise_comb2 is the best model under all the different types of noises used. From experiments in previous sections it can be observed that no model was performing the best under all the noise types like this. Therefore these results show evidence that by training under noise and using both magnitude an MGD as inputs we can build SER models that are robust to many different types of indoor noises.

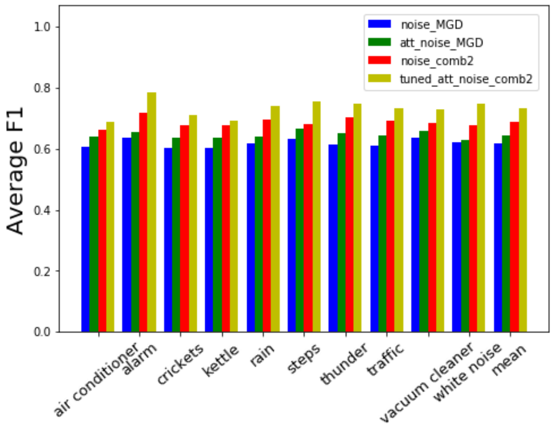

Figure 18: Improvement when under attention model

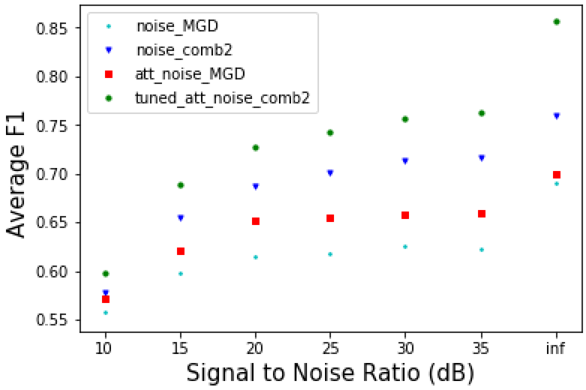

Figure 19: Mean performance under attention

4.4. *Incorporating attention mechanism*

To test the effectiveness of the attention mechanism, several models from the previous section were chosen and attention mechanism was added to them as described in section 3.10. Magnitude was not considered in this section because it was the worst performing model from previous section. All the models in this section are trained with noise as mentioned in the previous section. In addition to that, attention mechanism was incorporated. The models with attention mechanism are called att_noise_MGD and tuned_att_noise_comb2. These models use MGD and comb2 features respectively. tuned_att_noise_comb2 was fine tuned to increase the performance. During the fine tuning process, several filter sizes and number of filters were tried and the fine tuning was performed with the validation accuracy value of clean data. No such fine tuning was performed on att_noise_MGD. The reasoning behind just choosing the com2 model for fine tuning is that from the previous section it was seen that noise_comb2 performs the best under all different noise types and levels. Therefore it was assumed that com2 model will perform better with the added attention mechanism and fine tuning.

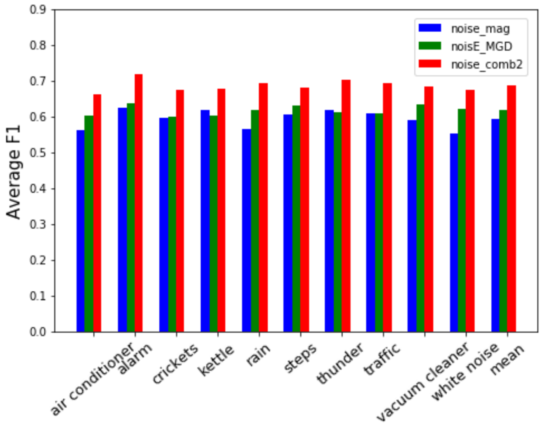

Figure 17: Performance when training under noise

From Figure 18 it can be seen that the average F1 score of MGD model improves from 0.61 to 0.64 after adding attention mechanism. tuned_noise_att_comb2 performs the best. This comb2 model improved from 0.68 to 0.73. From Figure 19 it can be seen that for clean speech, this model shows a F1 score of 0.85.

Figure 20 shows the per emotion performance of tuned_noise_att_comb2. These values are taken by averaging the F1 scores over all the different SNR levels and noise types per each emotion.

This section provides evidence that incorporating attention mechanism to CNN based SER systems can improve its robustness to noise. This may be due to the ability of attention mechanism to focus on important sections of the input and ignore the rest.

Figure 21 compares the performance of the model tuned_att_noise_comb2 with the model W-WPCC from [7]. They use an importance-weighted support vector machine to classify features based on sub-band spectral centroid weighted wavelet packet cepstral coefficients. [7] evaluates their model only under AWGN conditions. Therefore Figure 21 shows the performance of both models only under AWGN. Our model performs better under both clean and noisy speech. Note these results use accuracy instead of F1 score because [7] only reports accuracy. For clean speech, tuned_att_noise_comb2 performed at an accuracy level 86% and W-WPCC was 73%. Under speech of SNR of 20,

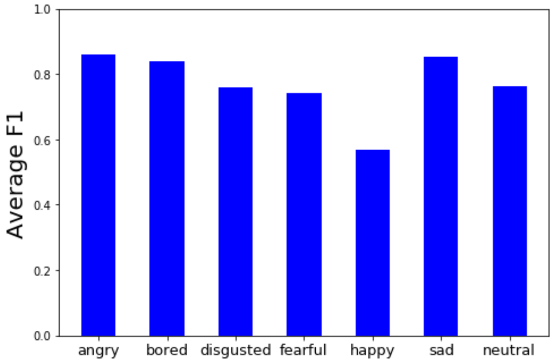

Figure 20: Per emotion performance

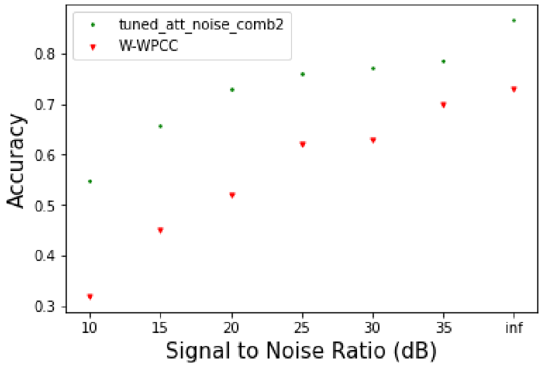

Figure 21: Comparison with W-WPCC model from [7]

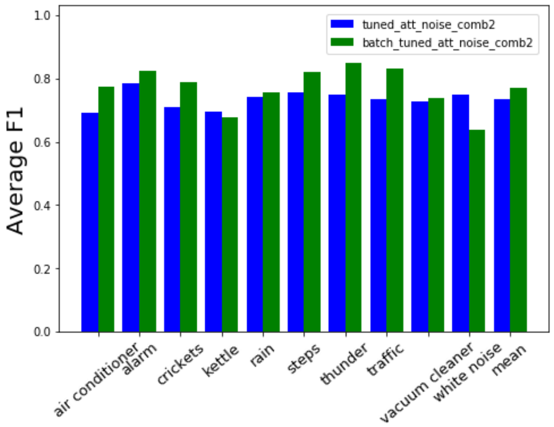

Figure 22: Comparison of batch training and instance wise training

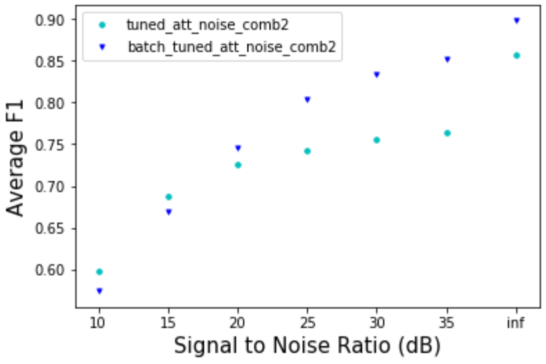

Figure 23: Comparison of batch training and instance wise training

our model performed at 72% and W-WPCC performed at 52%. If the average performance under all the noise types, SNR from 15 to 35 and clean speech is considered, W-WPCC performs at accuracy of 56% and our model performs at 76%. Note that [7] uses noise robust features. But they do not employ other techniques such as training with noise or attention mechanism.

4.5. Effects of batch-wise training

This section explains the effectiveness of batch training of the tuned_att_noise_comb2 model. Here the model batch_tuned_att_noise_comb2 was trained batch-wise. During earlier experiments, all the models were trained with batch size of 1 since Keras does not allow variable input sizes in the same batch even with FCNN. During batch training, for each batch the inputs were cropped into a randomly chosen length (along time axis). So all the samples in a batch had the same size. The evaluation procedure is the same as before. After a few trail-and-error experiments, a batch size of 16 was selected. Figure 22 and Figure 23 compares the F1 scores obtained via batch training and instance wise training. From Figure 23 it can be seen that the average performance of batch training under lower noise levels (SNR $\geq$ 20) is significantly better than that of the instance wise trained model. According to Figure 22 the mean performance improves from 0.73 to 0.76 when trained batch-wise. But interestingly the batch trained model performs worse than the instance trained model in certain noise conditions such as white noise and kettle noise. Batch trained model is much better than the instance model under some noise conditions such as thunder and traffic noise. Therefore this batch trained model may be useful under certain noise conditions.

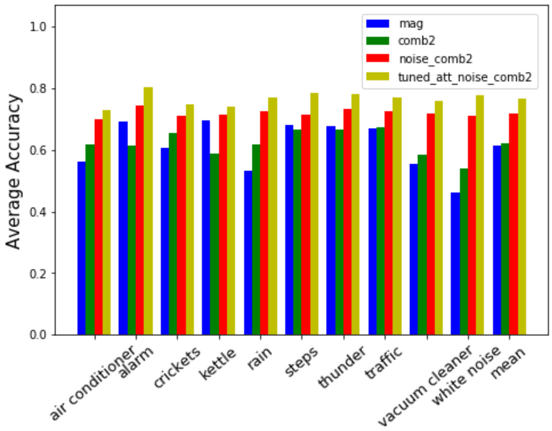

Figure 24: Improvement comparison of different methods

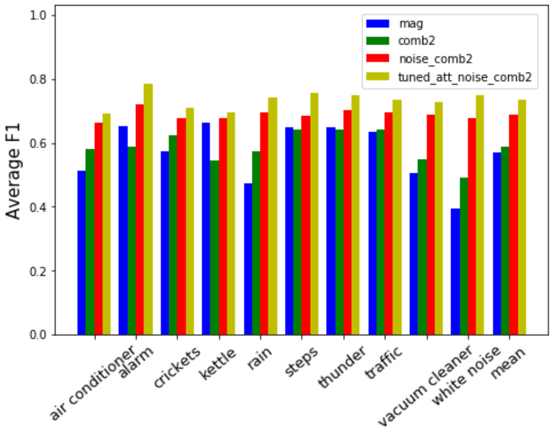

Figure 25: Improvement comparison of different methods

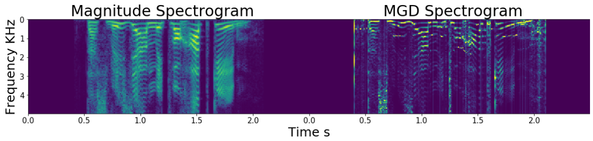

Figure 26: comb2 with silence

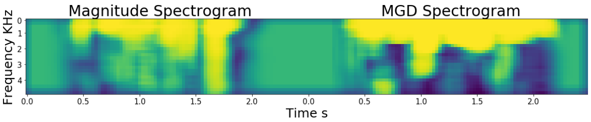

Figure 27: Attention map

4.6. Comparison of various methods on noise robustness

This section presents the effectiveness of various methods used to improve noise robustness and their comparison. Figure 24 shows the accuracy levels of the models trained with just using magnitude (mag), comb2, com2 model trained with noise and comb2 model with attention trained with noise. Taking mag model as a baseline, using combined features (mag and MGD) improved the accuracy by 1%. Using combined features and training the model with noise improved the accuracy by 10%. Using comb2, noise in training data and also incorporating attention mechanism improved the accuracy by 15%. From Figure 25, these value in F1 score are 0.01, 0.11 and 0.16. From these results it can be seen that combining magnitude and MGD spectrogram and training the model with noisy data have the biggest impact on building a noise robust model.

4.7. Operation of attention mechanism

Attention mechanism explicitly instructs the CNN model to focus only on important sections of the input features. This can be demonstrated by few examples. Figure 26 shows one example input to the classifier. There are silent sections in this speech clip. The silent sections are not important for classification of emotions. Figure 27 shows the attention map when the input is passed through the model tuned_att_noise _comb2. It can be seen that the attention mechanism has given less weights to the silent sections. Before training the models, the silent sections were removed from start and end of the speech data. Therefore, the model did not have an opportunity to experience silent sections. This might explain the model still paying some attention to silent sections.

Figure 28 shows another possible input to tuned_att_noise _comb2 model. In this input, some sections are hidden. These rectangular 'hidden' sections contains random noise. When this input is passed through the model, the generated attention map is shown in Figure 29. It can be seen that the model pays less attention to the the corrupted sections of the input. Furthermore, it can be seen that the MGD section does a better job of doing this. This gives evidence that the attention mechanism may be capable of handling data with missing/corrupted sections. For example, imagine a section of the input is corrupted with some noise. Certain noises are limited only to a certain range of frequencies and times as mentioned in Section 4.4. These types of noises might create artifacts which can be approximated by rectangular sections in Figure 28. Figure 29 provides evidence that attention mechanism may be able to handle situation similar to this.

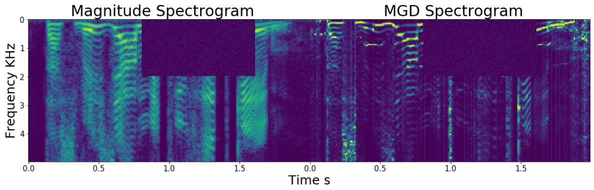

Figure 28: comb2 with corrupted sections

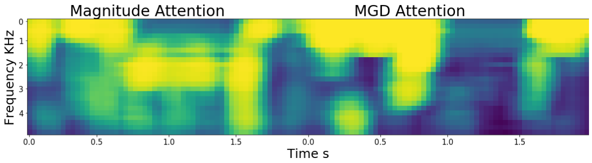

Figure 29: Attention map

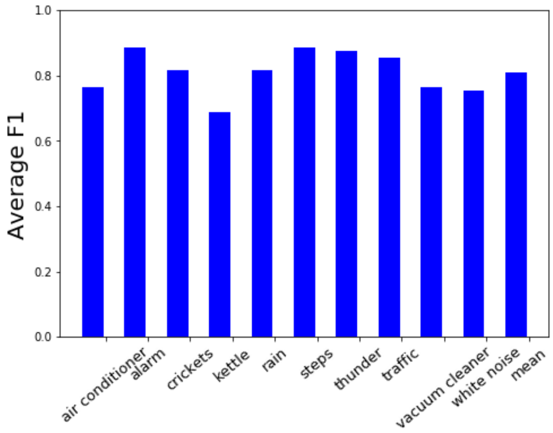

Figure 30: Per noise type performance on RAVDESS

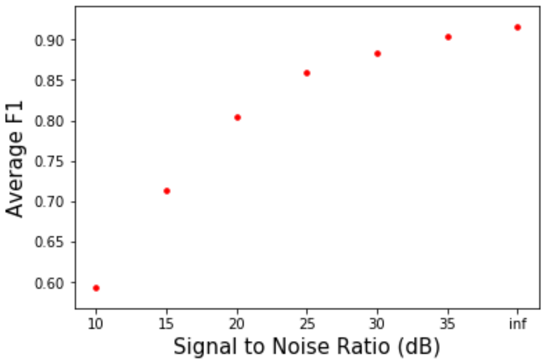

Figure 31: Average F1 on RAVDESS at various SNR levels

4.8. Evaluation on the RAVDESS dataset

The model in which hyperparameters were fine tuned on the Berlin dataset was used to train and evaluate on the RAVDESS dataset. The training was performed with batches as described in section 4.5. A subset of emotions were selected as mentioned in section 3.1. Since we used 6 emotions, the FCNN architecture was modified to accommodate that. Therefore, the last convolution layer of the FCNN had 6 filters instead of 7. All other hyperparameters were kept unchanged. Figure 30 shows the mean performance of the model under all the different SNR levels, but under different noise types. From this diagram it can be seen that the model has a mean F1 score of 0.81. This value is the performance of the system under all SNR levels and noise types considered. It can also be seen that kettle noise has the lowest level of performance which is 0.68. Alarms and steps noises have the highest performance which is 0.88. This is similar to the results from Berlin data set as can be seen from Section 4.4. This provides evidence that disturbances like alarms and step noises can be handled easily with our SER solution. But noises like kettle whistle noise are harder to handle for these systems. Figure 31 shows the model F1 values under different SNR values. These values were averaged over all the different noise types. The model shows F1 values of 0.91 under clean speech. Figure 32 shows the same evaluation results in terms of accuracy. It shows an accuracy of 91% under clean speech. Figure 33 shows an emotion wise breakdown of model F1 values. Here the average F1 was taken over all the SNR levels and noise types. From Figure 33 it can be seen that sad is the lowest performing emotion while calm performs best. Although the types of emotions used in training Berlin dataset if different from RAVDESS, certain common emotions are present in both datasets. The performance on these emotions differ in the two models. This may be due to the difference in the training data, difference in the emotions and randomness involved in the training process. Therefore, the we can conclude that the performance of each individual emotion may depend on these conditions and should be taken into consideration when using these models for practical applications.

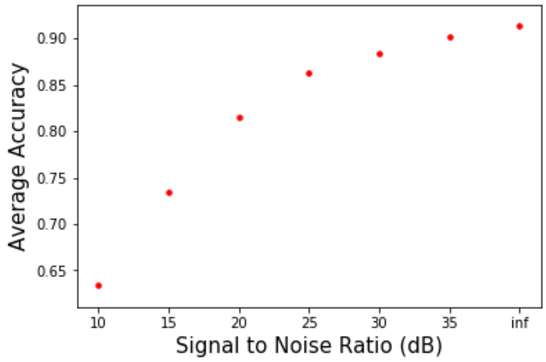

Figure 32: Average accuracy on RAVDESS at various SNR levels

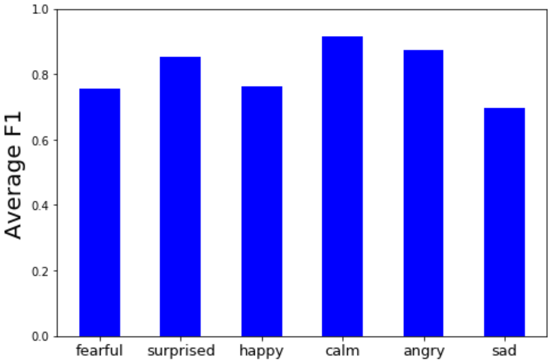

Figure 33: Per noise performance on RAVDESS

5. Discussion

From the results of this study it can be seen that the attention mechanism is capable of handling data with corrupted sections. But the training data consisted only of simple and artificial noise. If it contained other real noise types, the attention mechanism may have learned to deal with noise/corrupted data in a better manner. This hypothesis is yet to be tested.

The main motivation for using the attention mechanism is that it is capable of ignoring irrelevant sections of input and focus on the important sections. So, if the input consists of large sections of irrelevant data, the attention mechanism should pay less attention to this input. Thinking in this direction, it can be hypothesized that the attention layer by itself may be able to predict the uncertainty of the model. This has to be tested during future studies.

From figures 18 and 30 which explains the noise type performance of the models trained on Berlin and RAVDESS datasets, it can be seen that both of these models show their lowest performance under kettle noise. This may be evidence that SER systems created by the procedure we described are affected adversely by noises similar to the whistle sound of kettles. Figure 34 shows the kettle noise we used mixed with one of the speech samples from Berlin dataset. The time frequency characteristics of kettle noise are different from other noises in that the dominant frequency of the kettle noise shifts in time. But when our models were trained with noise, the artificial noises that were used only occupied a constant frequency ranges as can be seen from Figure 3. The low performance under kettle noise may be due to the fact that the models have not seen noises similar to this while training.

Referring to section 4.5 it can be seen that the performance of the model trained with batches decreased under white noise. This is in contrast to the performance under most of the other noise types where batch training improved performance over instance wise training. This may be due to certain characteristics of batch training. When training batch wise, the mean error it calculated per batch and then back propagated to update the weights. As explained in section 3.9 white noise was added to speech samples while training. Since white noise corrupts all the frequency ranges and time intervals in an equal manner, batch training might have cancelled the effect of white noise. Further analysis of this phenomena should be performed.

6. Conclusion

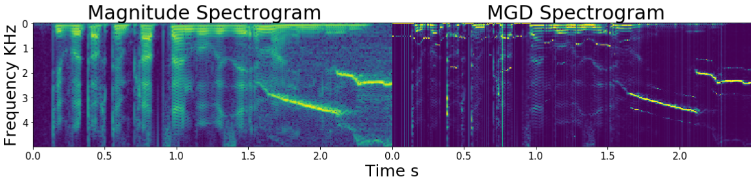

Figure 34: Speech with kettle noise

This study analyzes several ways to improve the performance of SER systems under noisy conditions. Magnitude spectrogram is the most common input feature for state-of-the-art CNN SER systems. But these are very sensitive to noise. We show that by combining magnitude spectrogram with modified group delay spectrogram as inputs to CNNs, the noise robustness of SER systems can be improved. Also it was observed that adding artificial noise during training can make models more robust to real world noises. Also, using CNN architecture characteristics like FCNN and attention mechanism can improve CNN based SER models. We first used the Berlin Database of Emotional Speech to find the noise robust features, training procedures, CNN architecture, and hyperparameters which perform well under noise. Then, we use these features, procedures, CNN architecture, and hyperparameters to train and evaluate a model on the speech section of the RAVDESS dataset.

Using the Berlin Database of Emotional Speech we showed that our final model with attention mechanism improved performance over other models considered. The 10 fold cross validation accuracy of the final model was 86% (F1 - 0.85) under clean speech and the average accuracy under all the noise types and signal to noise ratios considered was 76% (F1 - 0.73). The model trained on the speech section of RAVDESS dataset achieved an accuracy of 91% (F1 of 0.91) under clean speech and average accuracy of 82% (F1 of 0.81) under all signal to noise ratios and noise types considered. These results show that the noise robust features, training methods and the particular FCNN architecture with the attention mechanism that we obtained can indeed handle noisy data for the task of speech emotion recognition. Models trained and code used to produce results in this paper can be found at https://github.com/sleekEagle/noise_emotion.git

References

[1] C. E. Izard, Human emotions, Springer Science & Business Media, 2013.

[2] J. Zhao, X. Mao, L. Chen, Speech emotion recognition using deep 1D & 2D CNN LSTM networks, Biomedical Signal Processing and Control 47 (2019) 312–323. doi:10.1016/j.bspc.2018.08.035.

[3] A. M. Badshah, J. Ahmad, N. Rahim, S. W. Baik, Speech Emotion Recognition from Spectrograms with Deep Convolutional Neural Network, in: 2017 International Conference on Platform Technology and Service, PlatCon 2017 - Proceedings, IEEE, 2017, pp. 1–5. doi:10.1109/PlatCon.2017.7883728.

[4] W. Lim, D. Jang, T. Lee, Speech emotion recognition using convolutional and Recurrent Neural Networks, in: 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA 2016, IEEE, 2017, pp. 1–4. doi:10.1109/APSIPA.2016.7820699.

[5] J. Deng, X. Xu, Z. Zhang, S. Frühholz, D. Grandjean, B. Schuller, Fisher kernels on phase-based features for speech emotion recognition, in: Dialogues with social robots, Springer, 2017, pp. 195–203.

[6] scipy.signal.spectrogram.
URL `https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.spectrogram.html`

[7] Y. Huang, W. Ao, G. Zhang, Novel sub-band spectral centroid weighted wavelet packet features with importance-weighted support vector machines for robust speech emotion recognition, Wireless Personal Communications 95 (3) (2017) 2223–2238.

[8] 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 (5) (2018) e0196391.

[9] L. Wyse, Audio Spectrogram Representations for Processing with Convolutional Neural Networks, arXiv preprint arXiv:1706.09559 (2017). arXiv:1706.09559.
URL `http://arxiv.org/abs/1706.09559`

[10] H. M. Fayek, M. Lech, L. Cavedon, Evaluating deep learning architectures for Speech Emotion Recognition, Neural Networks 92 (2017) 60–68. doi:10.1016/j.neunet.2017.02.013.

[11] L. Hertel, H. Phan, A. Mertins, Comparing time and frequency domain for audio event recognition using deep learning, in: Proceedings of the International Joint Conference on Neural Networks, Vol. 2016-Octob, IEEE, 2016, pp. 3407–3411. doi:10.1109/IJCNN.2016.7727635.

[12] I. Ozer, Z. Ozer, O. Findik, Noise robust sound event classification with convolutional neural network, Neurocomputing 272 (2018) 505–512. doi:10.1016/j.neucom.2017.07.021.

[13] A. Satt, S. Rozenberg, R. Hoory, Efficient emotion recognition from speech using deep learning on spectrograms, in: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, Vol. 2017-Augus, 2017, pp. 1089–1093. doi:10.21437/Interspeech.2017-200.

[14] Y. Ma, Y. Hao, M. Chen, J. Chen, P. Lu, A. Košir, Audio-visual emotion fusion (AVEF): A deep efficient weighted approach, Information Fusion 46 (2019) 184–192. doi:10.1016/j.inffus.2018.06.003.

[15] Y. Su, K. Zhang, J. Wang, K. Madani, Environment sound classification using a two-stream CNN based on decision-level fusion, Sensors (Switzerland) 19 (7) (2019) 1733. doi:10.3390/s19071733.

[16] Z. Huang, M. Dong, Q. Mao, Y. Zhan, Speech emotion recognition using CNN, in: MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia, ACM, 2014, pp. 801–804. doi:10.1145/2647868.2654984.

[17] S. Amiriparian, M. Gerczuk, S. Ottl, N. Cummins, M. Freitag, S. Pugachevskiy, A. Baird, B. Schuller, Snore sound classification using image-based deep spectrum features, in: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, Vol. 2017-Augus, 2017, pp. 3512–3516. doi:10.21437/Interspeech.2017-434.

[18] K. Qian, Z. Ren, V. Pandit, Z. Yang, Z. Zhang, B. Schuller, Wavelets Revisited for the Classification of Acoustic Scenes, in: Detection and Classification of Acoustic Scenes and Events (DCASE), 2017, pp. 1–5.

[19] R. M. Hegde, H. A. Murthy, V. R. R. Gadde, Significance of the modified group delay feature in speech recognition, IEEE Transactions on Audio, Speech, and Language Processing 15 (1) (2007) 190–202.

[20] X. Huang, A. Acero, H.-W. Hon, R. Foreword By-Reddy, Spoken language processing: A guide to theory, algorithm, and system development, Prentice hall PTR, 2001.

[21] L. Guo, L. Wang, J. Dang, L. Zhang, H. Guan, X. Li, Speech emotion recognition by combining amplitude and phase information using convolutional neural network, in: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, Vol. 2018-Septe, 2018, pp. 1611–1615. doi:10.21437/Interspeech.2018-2156.

[22] I. Paraskevas, M. Rangoussi, Feature Extraction for Audio Classification of Gunshots Using the Hartley Transform, Open Journal of Acoustics 02 (03) (2012) 131–142. doi:10.4236/oja.2012.23015.

[23] P. Mowlaee, R. Saeidi, Y. Stylianou, Phase importance in speech processing applications, in: Fifteenth Annual Conference of the International Speech Communication Association, 2014.

[24] M. Dörfler, R. Bammer, T. Grill, Inside the spectrogram: Convolutional Neural Networks in audio processing, in: 2017 12th International Conference on Sampling Theory and Applications, SampTA 2017, IEEE, 2017, pp. 152–155. doi:10.1109/SAMPTA.2017.8024472.

[25] Y. Han, K. Lee, Acoustic scene classification using convolutional neural network and multiple-width frequency-delta data augmentation, arXiv preprint arXiv:1607.02383 (2016). arXiv:1607.02383.
URL http://arxiv.org/abs/1607.02383

[26] O. Abdel-Hamid, A. R. Mohamed, H. Jiang, L. Deng, G. Penn, D. Yu, Convolutional neural networks for speech recognition, IEEE Transactions on Audio, Speech and Language Processing 22 (10) (2014) 1533–1545. doi:10.1109/TASLP.2014.2339736.

[27] Y. Zhang, J. Du, Z. Wang, J. Zhang, Y. Tu, Attention based fully convolutional network for speech emotion recognition, in: 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), IEEE, 2018, pp. 1771–1775.

[28] M. Y. Ahmed, Z. Chen, E. Fass, J. Stankovic, Real time distant speech emotion recognition in indoor environments, in: Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, 2017, pp. 215–224.

[29] K. Kinoshita, M. Delcroix, S. Gannot, E. A. P. Habets, R. Haeb-Umbach, W. Kellermann, V. Leutnant, R. Maas, T. Nakatani, B. Raj, A summary of the REVERB challenge: state-of-the-art and remaining challenges in reverberant speech processing research, EURASIP Journal on Advances in Signal Processing 2016 (1) (2016) 7.

[30] R. F. Dickerson, E. Hoque, P. Asare, S. Nirjon, J. A. Stankovic, Resonate: reverberation environment simulation for improved classification of speech models, in: IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks, IEEE, 2014, pp. 107–117.

[31] A. Salekin, Z. Chen, M. Y. Ahmed, J. Lach, D. Metz, K. De La Haye, B. Bell, J. A. Stankovic, Distant emotion recognition, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 1 (3) (2017) 96.

[32] Y. Hu, P. C. Loizou, Evaluation of objective quality measures for speech enhancement, IEEE Transactions on audio, speech, and language processing 16 (1) (2007) 229–238.

[33] M. Mimura, S. Sakai, T. Kawahara, Joint Optimization of Denoising Autoencoder and DNN Acoustic Model Based on Multi-Target Learning for Noisy Speech Recognition., in: Interspeech, 2016, pp. 3803–3807.

[34] A. Satt, S. Rozenberg, R. Hoory, Efficient Emotion Recognition from Speech Using Deep Learning on Spectrograms., in: INTERSPEECH, 2017, pp. 1089–1093.

[35] K. Kumar, C. Kim, R. M. Stern, Delta-spectral cepstral coefficients for robust speech recognition, in: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, IEEE, 2011, pp. 4784–4787. doi:10.1109/ICASSP.2011.5947425.

[36] S. H. K. Parthasarathi, R. Padmanabhan, H. A. Murthy, Robustness of group delay representations for noisy speech signals, International Journal of Speech Technology 14 (4) (2011) 361.

[37] P. Rajan, S. H. K. Parthasarathi, H. A. Murthy, Robustness of phase based features for speaker recognition, Tech. rep. (2009).

[38] C.-W. Huang, S. S. Narayanan, Deep convolutional recurrent neural network with attention mechanism for robust speech emotion recognition, in: 2017 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2017, pp. 583–588.

[39] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, T.-S. Chua, Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 5659–5667.

[40] F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, B. Weiss, A database of German emotional speech, in: Ninth European Conference on Speech Communication and Technology, 2005.

[41] E. Fonseca, J. Pons Puig, X. Favory, F. Font Corbera, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, X. Serra, Freesound datasets: a platform for the creation of open audio datasets, in: Hu X, Cunningham SJ, Turnbull D, Duan Z, editors. Proceedings of the 18th ISMIR Conference; 2017 oct 23-27; Suzhou, China.[Canada]: International Society for Music Information Retrieval; 2017. p. 486-93., International Society for Music Information Retrieval (ISMIR), 2017.

[42] C. Huang, G. Chen, H. Yu, Y. Bao, L. Zhao, Speech emotion recognition under white noise, Archives of Acoustics 38 (4) (2013) 457–463.

[43] H. Jin, Q. Song, X. Hu, Auto-keras: An efficient neural architecture search system, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 1946–1956.