Audio processing methods for speech emotion recognition using machine learning

Oleksii Turuta[1,†], Iryna Afanasieva[1,†], Nataliia Golian[1,†], Vira Golian[1,†], Kostiantyn Onyshchenko[1,†] and Daniil Suvorov[1,*,†]

[1] *Kharkiv National University of Radio Electronics, Nauky Ave. 14, 61166, Kharkiv, Ukraine*

Abstract

The emotional state of an individual can be perceived and analyzed in a variety of ways. One such method is the analysis of a person's voice. In this analysis, the language of the person being analyzed does not matter; only the characteristic of the sound is important. This paper presents a research that utilizes artificial intelligence techniques, such as neural networks, to recognize emotion. The neural networks employed in this research include BiLSTM, GRU, CNN and CRNN. CREMA-D and IEMOCAP datasets, utilizing four emotions (anger, happiness, sadness and neutral), were employed in the research. MFCCs were used. Furthermore, the impact of augmentation (specifically, the addition of noise, time stretch and pitch shift) on the efficacy of the methodologies was analyzed. The research indicates that CNN is the most efficacious approach, with an accuracy of 98.8%. This result represents the most optimal outcome among comparable studies.

Keywords

audio, emotions, machine learning, speech, neural networks, recognition, artificial intelligence, python, tensorflow

1. Introduction

At present, the field of artificial intelligence is actively developing and is involved in various spheres of human activity. A large part of researches using approaches and methods of artificial intelligence is connected with human analysis and cognition. A considerable part of researches is connected with processing of human emotional state. Such works are related to the issues of determining the emotional coloring of text, handwriting, photos and audio sequences.

One such analysis task is speech emotion recognition (also known as SER). The results of such research can help to understand the peculiarities of this field and to create appropriate solutions that could be used in various spheres of life, ranging from smart house systems to medical applications, emergency services, and so on.

MoMLeT-2024: 6th International Workshop on Modern Machine Learning Technologies, May, 31 - June, 1, 2024, Lviv-Shatsk, Ukraine

[*] Corresponding author.

[†] These authors contributed equally.

oleksii.turuta@nure.ua (O. Turuta); iryna.afanasieva@nure.ua (I. Afanasieva); nataliia.golian@nure.ua (N. Golian); vira.golan@nure.ua (V. Golian); kostiantyn.onyshchenko@nure.ua (K. Onyshchenko); daniil.suvorov@nure.ua (D. Suvorov)

0000-0002-0970-8617 (O. Turuta); 0000-0003-4061-0332 (I. Afanasieva); 0000-0002-1390-3116 (N. Golian); 0000-0002-7196-5286 (V. Golian); 0000-0002-7746-4570 (K. Onyshchenko); 0009-0008-0083-1978 (D. Suvorov)

The purpose of this research is to find the most effective artificial intelligence method to solve SER problem. For this purpose, the following tasks should be solved.

- To analyze the results of research in the area of SER
- To draw conclusions on the most relevant and effective methods described in previous works
- To study datasets for this task that have been generated by third parties and select the most appropriate for the current research
- To study preprocessing approaches for audio
- To investigate the possibilities of data augmentation
- To identify the necessary metrics to compare the performance of the models to be developed
- To develop models based on the analysis of previous work
- To conduct a series of experiments in which to obtain a description of each model through metrics
- To analyze the results obtained

As a result, the paper will take into account the experience of similar works as much as possible, highlight the most important aspects of SER task, try to avoid the shortcomings of previous works, and describe in details the steps and approaches to solving the task of speech emotion recognition.

2. Related works

This domain attracts the attention of researchers on an annual basis. A significant number of works are exploring the potential applications of artificial intelligence in emotion recognition, with a particular focus on audio. The accelerating development of machine learning and deep learning technologies has led to the emergence of a range of tools that can be employed to address SER problem in a more elegant and efficient manner.

A considerable number of works utilize datasets that are entirely distinct from one another. The most commonly used datasets are RAVDESS and TESS. However, there are already a number of more representative datasets that have been identified by researchers.

The first significant study is "Emotional Speech Recognition Using Deep Neural Networks" 2022 [1]. This study employs a sizable IEMOCAP dataset comprising 10 emotions, which exhibit varying degrees of uniformity across the dataset. The authors achieved an accuracy of 97.54% for a model utilizing GRU for four emotions. Additionally, CNN and CRNN-based models were tested, achieving accuracies of 96.96% and 97.18%, respectively. In their work, the researchers used data augmentation techniques, including the addition of noise and shifting of formants. The positive impact of augmenting the data was demonstrated. The methodology employed a matrix approach, whereby the data obtained from the audio was fed to the input of the model in the form of a matrix. Both MFCCs (Mel-Frequency Cepstral Coefficients) and other parameters, such as spectral features, were utilized as audio features in the experiments.

The dataiku platform blog [2] provides an illustrative example of audio data processing with model training, showcasing the capabilities of the dataiku platform. However, the post delves into the specifics of the data preprocessing methodology. Additionally, the utilization of a combination of CREMA-D, RAVDESS, SAVEE, and TESS datasets for SER task is a strength that will undoubtedly enhance the objectivity of the resulting model. This is because the datasets are created by completely different groups of people who used different actors and approaches to record the voices. However, the paper only mentions the possibility of data augmentation, and the recognition accuracies range from 43% to 72% for six classes. The authors employed a combination of MFCCs and Mel-spectrograms as audio features.

In 2020, the authors of "Speech Emotion Recognition with deep learning" [3] employed an Auto-Encoder and SVM on the Ryerson Multimedia Laboratory dataset to address the challenge of emotion recognition from audio. MFCCs, Zero Crossing Rate, and other audio features were utilized in the experiments, resulting in an accuracy range of 65% to 74%, contingent on the model configuration.

The authors of the publication "Speech Emotion Recognition Using Deep Learning Techniques: A Review" [4] have created a review material that describes approaches to solving SER and the results of these approaches as of 2016. The publication describes the peculiarities of the order of audio processing for emotion classification and provides comparative characteristics of different emotions. At that time, deep learning demonstrated superior results, indicating the relevance of using neural networks for this task.

Another noteworthy study from 2023 is "A Deep Learning Approach for Speech Emotion Recognition Optimization Using Meta-Learning" [5]. In their publication, the authors present an extensive study of different datasets, their combinations, the impact of augmentations on learning, and the application of so-called meta-learning. Meta-learning encompasses the search for an optimizer and hyperparameter of the learning rate, the search for the most effective type of augmentation, and a set of audio features.

This work employed a vector-based approach, whereby extracted parameters from audio were converted from a matrix to a vector. This approach enabled the authors to achieve 83% and 91% accuracy for CREMA-D and RAVDESS+TESS+SAVEE+CREMA-D datasets, respectively. It was also observed that data augmentation using audio time stretching has the greatest positive effect on accuracy, while the other types of augmentation show less qualitative results. However, some datasets in this study show 100% accuracy, which raises questions as to how representative and variable such datasets are. Additionally, this paper analyzes only the CNN-based model and its combination with multiple LSTM layers.

Another study, "Speech-Based Emotion Recognition", published in the "International Journal for Research" [6], explores the application of convolutional neural networks for emotion recognition. The paper utilizes RAVDESS dataset, comprising five emotions, and a convolutional layer based neural network (six layers) with a 1D convolutional layers. Additionally, the authors employ MFCCs as audio features, which serve as the input to the model. The authors assert that their approach yields high accuracy. The f1-score was employed as a metric, resulting in a value of 0.91. Additionally, the authors proposed that the outcomes of such studies could be integrated into a recommendation algorithm for marketplaces, which represents a promising application that warrants further investigation.

3. Methods and Materials

3.1. Datasets

At the time of this research, there are a sufficient number of datasets on the Web that have been created to find a solution to SER problem. Each of the datasets has its own structure and characteristics in terms of phrases, emotions, actors who speak the phrase, level of emotionality, and so on.

Let's look at the most popular and interesting datasets that can be found in the network and that have been used by previous works.

- SAVEE
- RAVDESS
- TESS
- CREMA-D
- IEMOCAP

Let's take a closer look at each of these datasets.

3.1.1. SAVEE

SAVEE (Surrey Audio-Visual Expressed Emotion) [7] is a dataset created for research in emotion recognition using audio and visual information. The main focus of SAVEE is the recognition of expressed emotion in speech.

The main features of the SAVEE dataset are the following.

- SAVEE contains audio recordings of four men reading sentences expressing four different emotions: happiness, sadness, anger and fear. Each participant reads 15 sentences for each emotion
- SAVEE focuses on the four basic emotions, providing a balance between positive (happiness) and negative (sadness, anger, fear) emotional states

3.1.2. RAVDESS

RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) [8] is a dataset designed to study emotion recognition using audio and visual information, including audio and video recordings.

The main features of the RAVDESS dataset are the following.

- The dataset contains audio and video recordings of professional actors performing phrases with different emotional expressions. In total, the dataset contains about 1,440 high quality files
- RAVDESS includes 6 emotions including happiness, anger, fear, disgust, sadness and neutral. Each actor performs phrases for each emotion
- There are 12 male and 12 female actors

3.1.3. TESS

TESS (Toronto Emotional Speech Set) [9] is a dataset of audio recordings of emotional speech created for the research and development of systems for recognizing emotions from audio information. The dataset consists of audio recordings of women and men expressing different emotions.

The main features of TESS are the following.

- The set includes audio recordings of speech performed by English-speaking actors expressing different emotions. In total, the set contains more than 2,800 audio files
- TESS contains six basic emotions: happiness, sadness, anger, fear, disgust and neutral. Each audio file represents one of these emotions
- Each audio file also has information about the intensity of the emotional expression

3.1.4. CREMA-D

CREMA-D (Crowd-sourced Emotional Multimodal Actors Dataset) [10] is a dataset created to study and develop emotion recognition systems using audio and visual information. This dataset is unique in that it combines audio and video recordings of professional actors reciting sentences with different emotional expressions.

Key features of CREMA-D are the following.

- The dataset contains audio and video recordings of over 90 professional actors reading phrases designed to express a variety of emotions. The phrases are specially designed for this dataset
- CREMA-D covers a wide range of emotions such as happiness, sadness, anger, disgust, surprise and neutral state
- Each actor recites 12 phrases in three repetitions to provide variety in the data. The total number of audio files in the dataset is over 7,000

3.1.5. IEMOCAP

IEMOCAP (Interactive Emotional Dyadic Motion Capture) [11] is a dataset designed for research in the detection and analysis of emotional states in audio- and video-based communication. This dataset is unique in that it contains interactive communication sessions between actors reflecting different emotional states.

The main features of the IEMOCAP dataset are the following.

- The dataset contains audio and video recordings of interactions between actors in specially prepared and randomized scenarios
- IEMOCAP covers a wide range of emotions, including happiness, sadness, anger, fear, surprise, disgust, neutral state, and so on
- Each session contains dialogues between pairs of actors that replicate real-life situations, discussions, debates, or simple communication
- The total number of sessions in the set is more than 1,000 with a total audio duration of more than 12 hours, taking into account the different actors and emotional states

3.1.6. Comparison

Some of the presented datasets contain not only audio but also visual information, which would allow recognition not only from a person's speech but also from the image of his face, for example. However, this research is limited to emotional state recognition from audio information only.

So, after a superficial description of each dataset, we can compare the features of each dataset in more detail. Comparative features of the most popular sets are shown in Table 1.

Table 1
Datasets comparison

		SAVEE	RAVDESS	TESS	CREMA-D	IEMOCAP
Total samples		480	1,440	2,800	7,442	10,039
Emotions	Anger	+	+	+	+	+
	Happiness	+	+	+	+	+
	Disgust	+	+	+	+	+
	Fear	+	+	+	+	+
	Sadness	+	+	+	+	+
	Surprise	+	+	+		+
	Neutral	+	+	+	+	+
	Calmness		+			
	Frustration					+
	Excitation					+
Total emotions		7	8	7	6	10
Text variations		15	2	20	12	A lot of
Samples per emotion		~60	195 (96 for neutral)	400	~1,270	Non uniform
Speakers		4 (M)	24 (12 M/12F)	2 (F)	91 (48M/43F)	10 (5M/5F)
Emotion levels		1	2	1	4	A lot of

As can be seen from the table above, the most interesting datasets are CREMA-D and IEMOCAP, because they have a relatively large number of samples, as well as a large variation of text and actors. In addition, CREMA-D has up to 4 levels of emotionality, and IEMOCAP has a huge number of text variations that are not just read, but spoken as in real life (i.e. improvised recordings, which is important for creating more objective mathematical model).

The disadvantage of IEMOCAP dataset is a rather uneven distribution of classes, as can be seen in Figure 1.

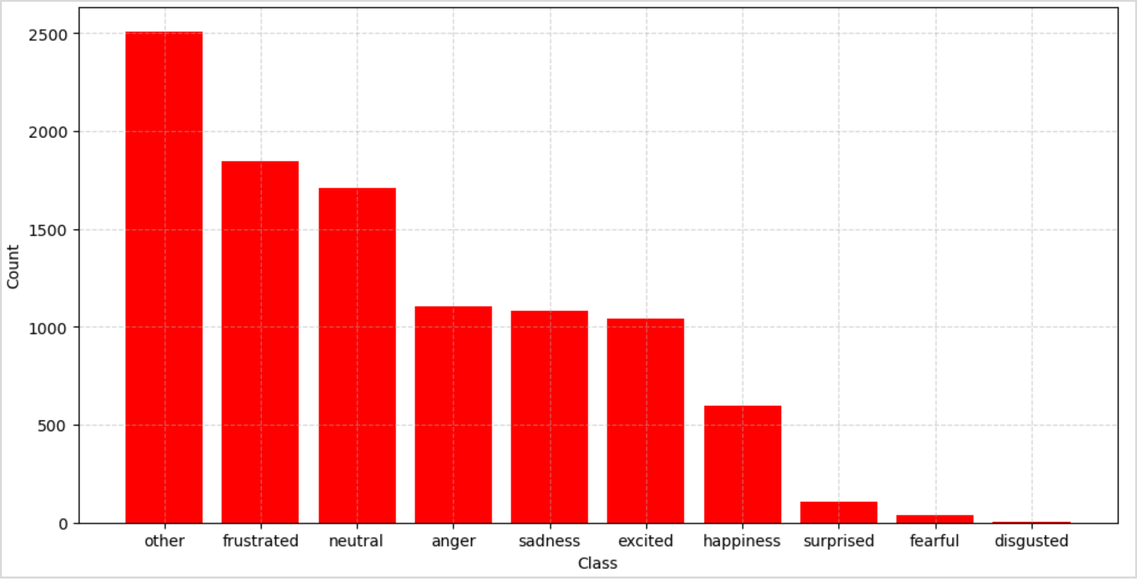

Figure 1: Classes distribution in IEMOCAP dataset

Since CREMA-D and IEMOCAP datasets are the most applicable from a theoretical point of view, these two will be used for further work. And in order to compare the models trained on these datasets as objectively as possible, only some emotions will be chosen. This way, the number and distribution of class instances in both sets will be approximately the same. Thus, the following emotions will be considered in this work.

- Anger
- Sadness
- Happiness
- Neutral

As a result, it will be possible to say how the configuration of the dataset affects the performance of the model.

3.2. Audio features

Artificial intelligence methods used in the experiments work with numerical data, i.e. in order for the model to recognize emotion in sound, it must be given a sequence of numbers as input, which would be a representation of the sound and contain data on the basis of which it is possible to make an assumption about the emotion. In the case of images and video, such data is the pixel values in each of the RGB channels. In the case of sound, obtaining numerical values is a bit more complex, although it is fairly obvious.

Before we look at the algorithm for extracting audio features, let's consider how an audio signal is represented digitally and what audio features can be [12].

3.2.1. Digital representation of audio

In usual (analog) world, sound represents continuous waves of a particular frequency (or, more often, a set of frequencies). A familiar representation of a sound wave is shown in Figure 2.

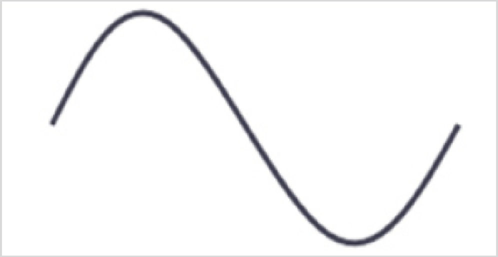

Figure 2: Analog sound wave

In analog-to-digital conversion (ADC), the analog signal is read by a recorder at a certain interval, also known as sample rate, which is measured in Hz. Figure 3 shows what the signal looks like after ADC.

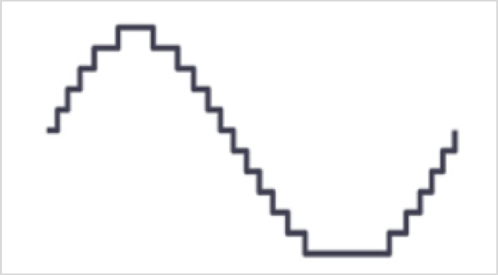

Figure 3: Digital sound wave

Now it is not continuous and consists of samples. It is in this form that the computer can work with and process the signal. Logically, the higher the sample rate, the more samples there are in the digital representation of the audio, and the more detail can be extracted from the analog signal. Often 44,100 Hz is used as the sample rate value. When talking about audio processing with artificial intelligence, it is often limited to 22,050 Hz.

3.2.2. Features types

When processing audio data (whether speech or just sounds, such as urban sounds), three types of features are distinguished.

- Time domain
- Frequency domain
- Time-frequency domain

Time domain features are signal characteristics that are directly related to the amplitude or strength of the signal as a function of time. These features provide information about how a signal changes with time without regard to its frequency components.

Frequency domain features are signal features that are related to the frequency composition of the signal. Rather than analyzing how a signal changes with time, these features provide information about the distribution of energy among the various frequencies present in the whole signal.

Time-frequency domain features are signal features that describe the frequency changes in a signal over time (usually represented as spectrograms).

Let us look at some of the more common time domain features.

Amplitude envelope shows the maximum amplitude among samples of a particular section of audio. Gives you a rough idea of the volume of the signal, and can also help determine the beginning of an event (the beginning of a note, the beginning of a word, and so on).

RMS (Root Mean Square) – just the root mean square of all samples for an audio segment also provides information about the loudness of the signal and is less sensitive to outliers (volume peaks). In the context of machine learning, the RMS can help in audio segmentation, i.e., distinguishing between signal fragments.

ZCR (Zero Crossing Rate) shows how many times a signal crosses the zero axis in a given time. This feature is also very commonly used and can be useful in determining when there is a transition between phonemes or words in speech. A change in the ZCR can indicate boundaries or changes in the acoustic properties of speech.

Let's look at some frequency domain features that can be used for machine learning.

Amplitude spectrum shows the amplitudes of the different frequencies present in the signal.

Spectral centroid represents the "center of mass" of a signal's frequency spectrum and indicates where the average frequency value is; a high spectral centroid may indicate high frequency content, while a low centroid may indicate low frequency content.

Spectral bandwidth defines the width of the frequency range.

Spectral flatness measures how evenly energy is distributed across the frequency spectrum; a value near 1 indicates a more uniform spectrum, while a value near 0 indicates focused frequency content.

Spectral rolloff indicates the frequency below which a certain percentage of the spectrum's energy is located (typically 85%); a high spectral rolloff can indicate that most of the signal's energy is concentrated in the higher frequencies.

Time-frequency domain features are perhaps the most interesting and contain much more information about the sound. They are the ones that allow you to map the features of a sound over time. Let's take a look at the basic time-frequency domain features.

A spectrogram is a visual representation of the frequency spectrum of a signal as it changes with time. It is the simplest representation of time-frequency domain features. However, there are more descriptive and closer to human perception representations of such features.

The Mel-spectrogram is a type of spectrogram used in audio and speech processing. It takes into account the peculiarities of sound perception by the human ear and uses the Mel scale to spatially represent frequencies.

Examples of regular and Mel-spectrograms are shown in Figures 4 and 5.

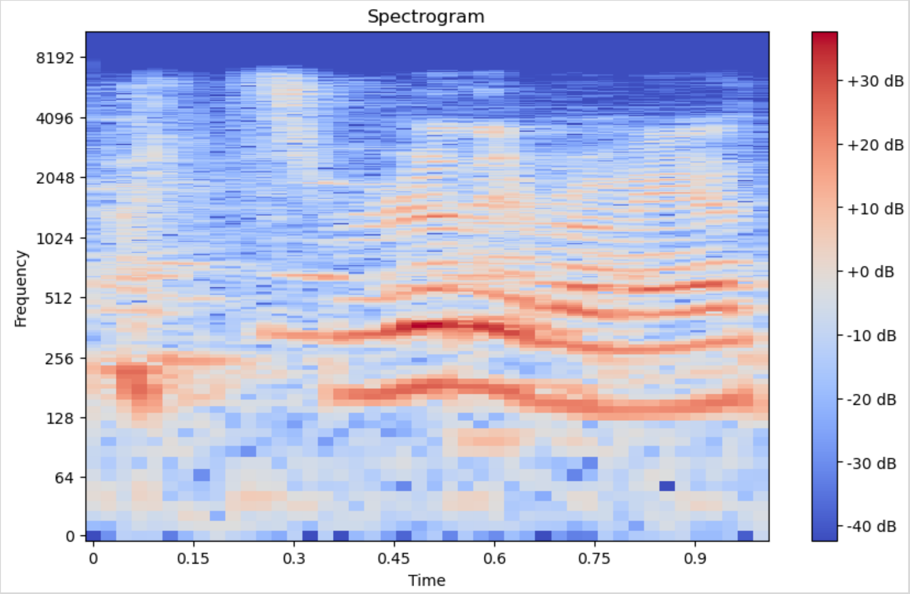

Figure 4: Example of regular spectrogram

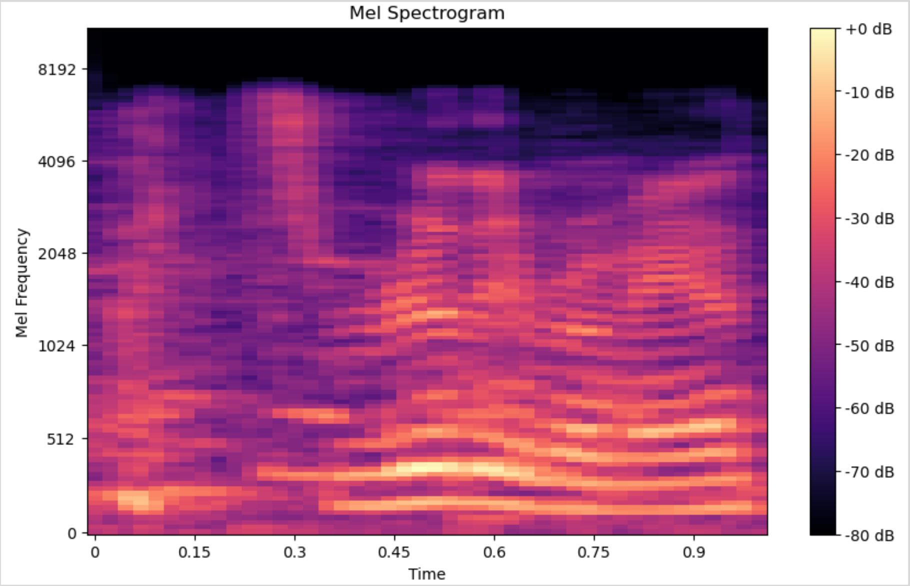

Figure 5: Example of Mel-spectrogram

Obvious difference between these two views for the same audio segment.

MFCCs (Mel-Frequency Cepstral Coefficients) are cepstral mapping coefficients used to describe sound signals, particularly in speech processing and speech recognition. They result from an attempt to model the perception of sound by the human ear and the properties of speech. The detailed derivation of these coefficients is described in extraction section. However, the coefficients are also visually represented as a spectrogram, as shown in Figure 6.

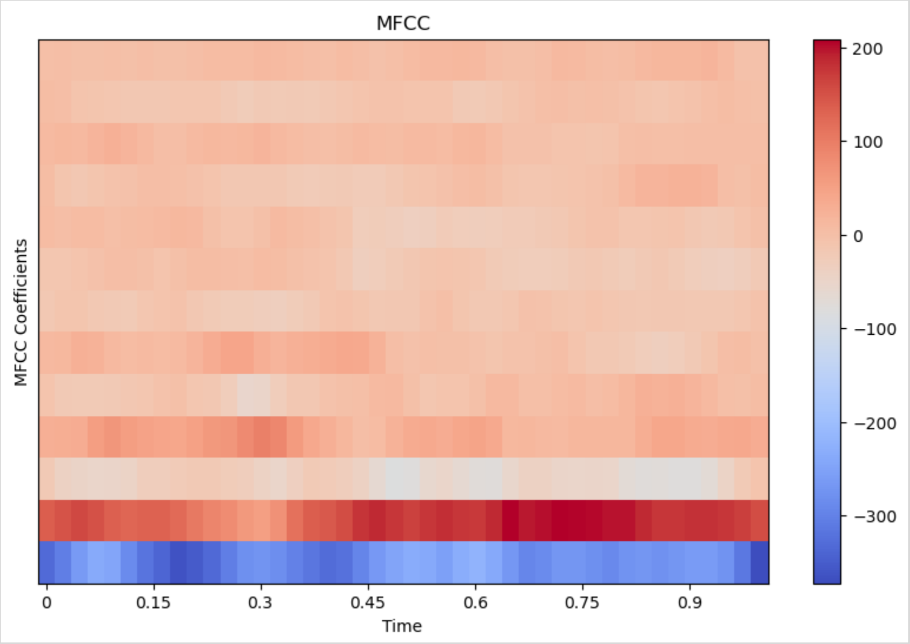

Figure 6: Example of MFCCs representation

The number of coefficients is not limited and is chosen by the researcher. As can be seen from the spectrogram (it contains 13 coefficients), most of the initial coefficients are representative. Therefore, a large number of coefficients may slow down learning rather than improve accuracy.

MFCCs delta and delta-delta are also sometimes used in audio processing. These are nothing more than a derivative of first- and second-order MFCCs. They are used to capture the time dynamics of an audio signal and provide information about how the MFCCs change over time.

Based on the works analyzed, the highest efficiency is achieved by models using MFCCs. This is sufficient to achieve relatively high accuracy. Therefore, MFCCs are used in this paper as the most representative type of features for audio analysis.

3.2.3. Features extraction

The extraction of audio features follows several steps and differs slightly for different types of features.

However, the first step for each type is to divide the audio signal into frames.

Since an audio signal is a rather long sequence of samples, it makes sense to divide this sequence into subsequences and work with each of them. This is a fairly common practice in machine learning, which is also used in audio processing.

So the first step is to divide the signal into frames, each of which contains a certain number of samples. However, if you divide the signal into frames that are not related in any way, you may lose information about signal changes between frames. This is where overlapping comes in. This means that each subsequent frame overlaps the previous frame by a certain number of samples. Thus, both frames have common information at the transition point. This can also be visualized in another way. If the size of the frames is the same, 2048 samples, and we imagine that the beginning of the next frame is 512 samples away from the beginning of the previous frame, then the overlap (the number of samples contained in both frames) is 1536 samples. So, with this splitting of the audio signal, the frame size is 2048, and these 512 offset samples are called the hop size. This process is shown in Figure 7.

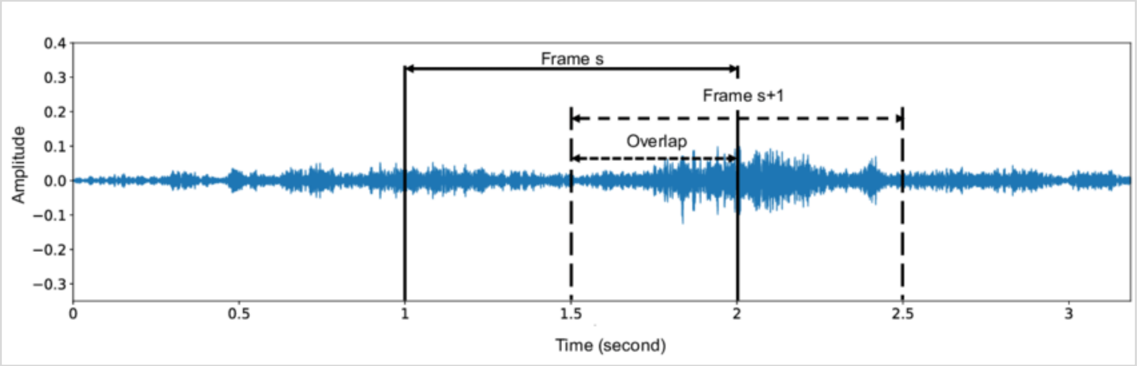

Figure 7: Framing and overlapping

Thus, after the first step, we have a set of frames connected by overlapping.

In the case of time-domain features, the next step is to extract features (amplitude envelope, RMS, etc.) from each frame. The numerical values after extraction are the input data for training the mathematical models.

In the case of frequency and time-frequency domains, there are two additional steps before features extraction.

The first step for both frequency and time-frequency domain features is to apply a windowing function to each frame. One of the most popular such functions is the Hamming window. The purpose of this function is described below.

So what does windowing do? It "smooths" the sound signal at the edges of the frame. And thanks to overlapping, data on these parts are not lost and are processed by the model in the same way. The application of the windowing function is shown in Figure 8.

What is the purpose of applying this function?

The next step in preparing frames for frequency domain and time-frequency domain features is to apply the Fourier Transform (FT) or the Short Time Fourier Transform (STFT) to the frames.

The Fourier Transform is a mathematical tool used to transform a signal from the time domain to the frequency domain. This transformation determines which frequencies make up a given signal and what their amplitude is. This is what windowing is used for, because without it, spectral leakage is observed.

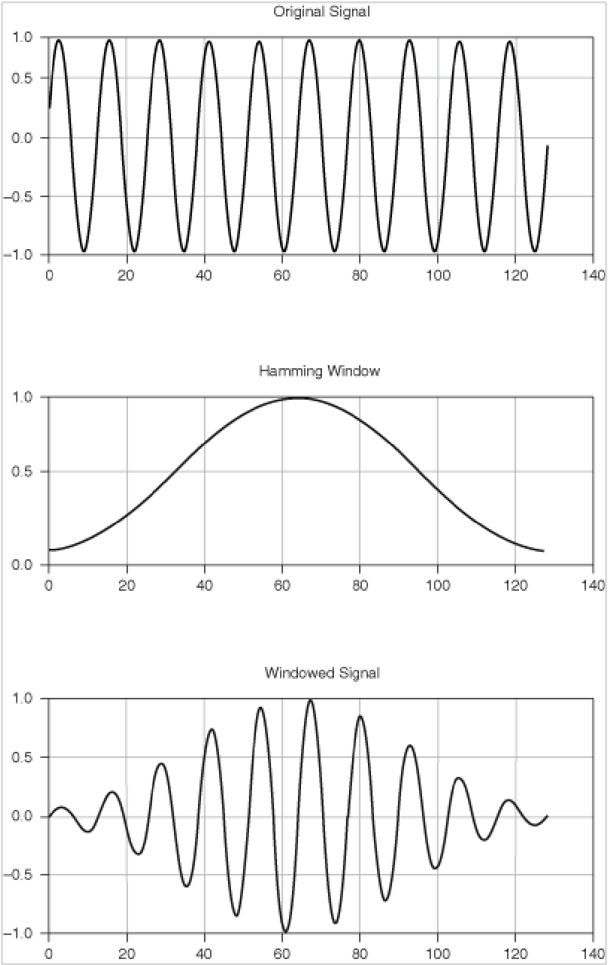

Figure 8: Windowing

An example of the conversion result without applying the windowing function is shown in Figure 9.

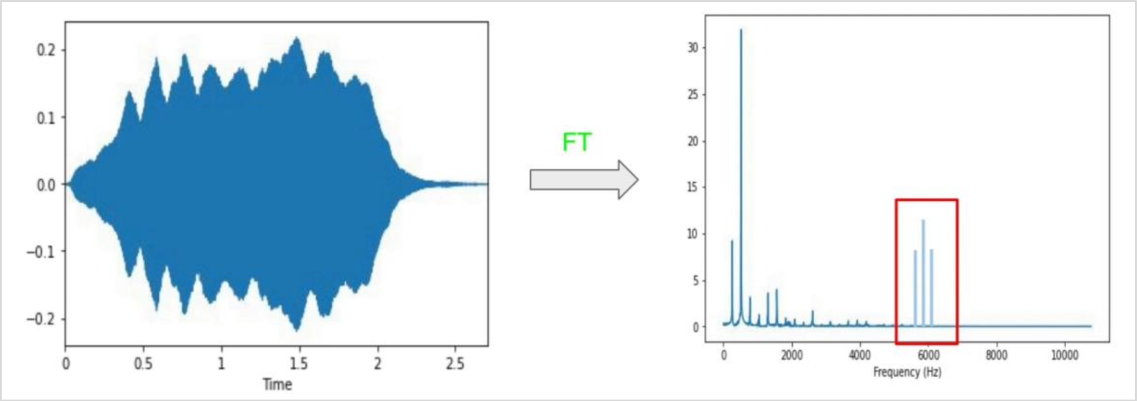

Figure 9: Spectral leakage representation after applying FT without windowing

The same effect has a windowing effect on the STFT. The STFT also transforms the signal from the time domain to the time-frequency domain, resulting in a three-dimensional representation of the signal, where one dimension is frequency, another is time, and the third is the amplitude of the signal at each point. This representation, assembled into an image, is called a spectrogram. After this transformation, the features can be extracted from the images.

For frequency domain, these are various spectral features, and for time-frequency domain features, these are spectrograms and MFCCs.

To obtain a Mel-spectrogram from a regular spectrogram obtained after applying the FT, Mel filters are applied to the regular spectrogram, which allows scaling the frequency scale to one more familiar to the human ear.

To obtain the Mel-Frequency Cepstral Coefficients, the logarithm of each Mel filter is taken from the Mel-spectrogram, and then the cepstral transformation is performed using the DCT (Discrete Cosine Transform) [13].

It is also important to note that when analyzing similar works, two approaches to preprocessing the input data have been noticed.

Since convolutional neural networks are often used in works, different authors have used both vector (1D) and matrix (2D) versions. For example, in the case of MFCCs, the audio is represented as a matrix rather than a vector. In such a case, the authors averaged the values of each coefficient across the audio frames, which resulted in a vector and used it as input to the convolutional vector layer model (1D).

In contrast, other authors did not use averaging and used a matrix as it is as input to the convolutional networks. This approach should be more effective, since averaging can largely mask important emotion information (this is evident in the analyzed papers: options with matrix input show much better results).

Another important point is the partitioning into frames. Often fixed values for frame size and hop size are used. However, since the length of audios varies, it will be impossible to divide them all into the same number of samples with fixed frame and hop sizes. So some audio will either not be processed completely or will be discarded altogether.

In this case it was decided to use dynamic frame and hop size. It is calculated from a fixed number of frames in the audio. This way, every data sample without exception will be used in the training.

3.2.4. Features extraction algorithm

Since we have determined that the most efficient features for SER task are MFCCs, the algorithm for extracting them is as follows.

1. Divide each audio file into frames. The frame size should be such that each audio file ends up consisting of 128 frames. The hop size will again be twice smaller than the frame size
2. Apply windowing to each frame
3. Apply FT to each frame
4. Convert the resulting spectrogram into a Mel-spectrogram
5. Convert the Mel-spectrogram into MFCCs via DCT with the number of coefficients 40 as the most optimal

6. Combine all frames of an audio into a two-dimensional matrix, which will be the input data for the mathematical model

Thus, each data sample will be converted into a matrix of numbers of size 128 by 40.

3.3. Data augmentation

A widely used data processing technique in machine learning is augmentation. In this technique, new data samples are created by applying various transformations to existing samples. This technique is used to increase the size of the training set and improve the overall ability of the model to generalize to new, real-world data.

The goals of augmentation include the following.

- Allowing the model to see more variation in the training set, which can help avoid overtraining and improve the generalization properties of the model
- Adding different variations to the data helps the model cope with different conditions and inputs
- Applying augmentation can help make the model less sensitive to changes in recording conditions or real-world scenarios

Data augmentation is an important step in building robust and efficient machine learning models. When talking about audio data augmentation, we can typically find data operations such as.

- Adding noise to audio
- Time-stretching
- Pitch shifting
- Applying various filters
- Formant shifting (changing the voice so that men's voices sound more feminine and women's voices sound more masculine)

Thus, augmenting the dataset with the above augmentation operations allows for a more objective and efficient mathematical model.

The most common types of augmentation chosen for this work are noise adding, time stretching (faster and slower), and pitch shifting (raising and lowering the pitch).

3.4. Processing methods

The most important stage is the choice of tools. In our case, the tools of artificial intelligence are methods of neural networks. Based on the analyzed works, the following methods have shown the most effective results.

- Convolutional neural networks
- Recurrent neural networks

Let's take a closer look at what each method is.

A convolutional neural network (CNN) is a type of deep neural network specifically designed to process and analyze structured matrices of data, such as images (although often used for images, such networks are also effective for audio processing because the audio signal can be represented graphically, such as a spectrogram).

In general, a convolutional network consists of several components, as shown in Figure 10.

- Convolution layer, which uses specific filters to perform the convolution operation
- Pooling layer, to reduce the matrix and highlight the most important features
- Fully connected layers (multiple layers of neurons, each with connections to each other)
- Activation layer

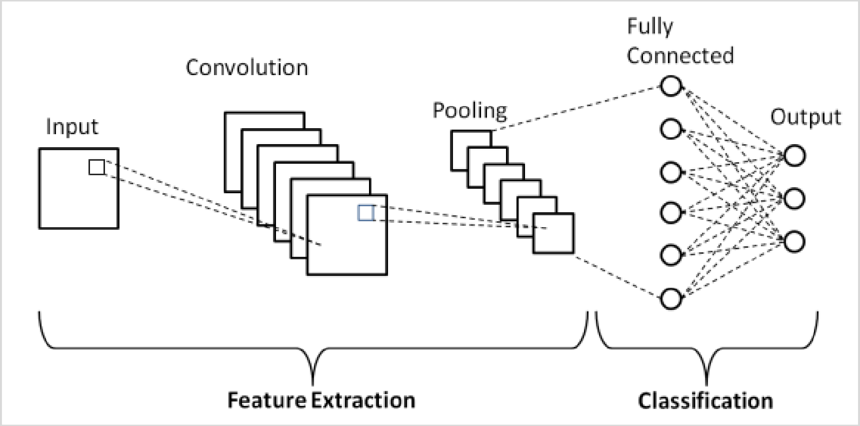

Figure 10: Simple structure of CNN

Thanks to filters in convolutional layers, which weights are adjusted during the network training process, the trained model is able to extract the necessary patterns (features) from the input data, which are further classified using a regular fully connected neural network.

RNN (Recurrent Neural Network) [14] is a class of neural networks designed to work with data sequences where information is distributed in time. The basic idea is that there are links in the network that create loops, allowing information from previous time steps to influence the current state of the network.

Among the recurrent networks, we can distinguish the main types.

- Regular recurrent network (RNN)
- Recurrent network with Long Short-Term Memory unit (LSTM)
- Recurrent network with a Gating Recurrent Unit (GRU)

A regular RNN has connections that form loops, allowing information from previous time steps to influence the current state of the network. Fast faces the problem of exploding or vanishing gradients when training on long sequences, which limits its ability to learn long-term dependencies. This type of recurrent network is not used in this paper as there are more efficient modifications.

Long Short-Term Memory (LSTM) [15] is an improvement of the RNN proposed in 1997 to overcome the limitations of the RNN short term memory and gradient problems.

The LSTM unit from which the recurrent network layer is built have three gates: an input gate, an output gate, and a forget gate. These gates control the flow of data needed to predict the output of the recurrent unit. The approximate structure of an LSTM unit is shown in Figure 11.

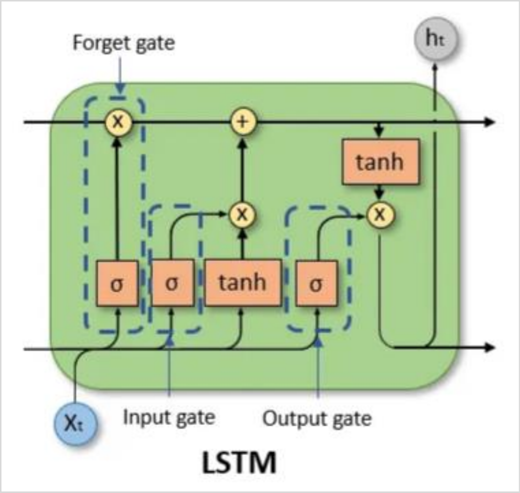

Figure 11: LSTM unit structure

GRU-based recurrent network [16] is similar to the LSTM in that it also helps solve the short-term memory of recurrent models. The GRU uses hidden states and has only two gates, a reset gate and an update gate. Similar to the LSTM, the reset and update gates control how much and what information is stored. An example architecture of a GRU is shown in Figure 12.

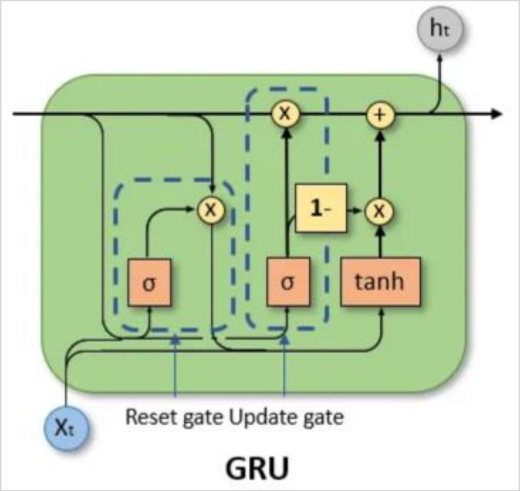

Figure 12: GRU structure

There are also modifications of recurrent networks called bidirectional RNN. This is a recurrent neural network architecture that works with data sequences in both directions (forward and backward).

The basic idea of bidirectional recurrent neural networks is that at each point in time, it computes predictions not only based on previous forward values, but also based on future values, which allows the model to participate in the context of both the past and the future [17].

In addition to the separate existence of these methods, it is common for machine learning to use a combination of them. For example, after extracting patterns through convolutional layers, a recurrent network is used for further processing.

Thus, the most interesting models for the research are based on the following methods are the following.

- BiLSTM
- GRU
- CNN
- CRNN (CNN + LSTM)

In this work, exactly these 4 types of models will be developed and studied.

3.5. Models architectures

The last layer of each network is a Fully Connected layer with 4 neurons and a softmax activation function to generate a probability for each of the 4 classes.

Batch Normalization and Dropout layers can be found in all models. Batch Normalization is used to speed up model training, the last to reduce model overtraining and has a value of 0.2 everywhere (the fraction of neurons/units that are turned off during training).

3.5.1. BiLSTM

The overall structure of the network is shown in Figure 13.

The first layer of the network has 128 units, the second 256 (however, as the layers are bidirectional, the number of units in each layer is doubled, i.e. 256 and 512).

These layers use the default tanh activation function. The Fully Connected layer of the network consists of 128 neurons, followed by a Softmax layer. The total number of parameters to be trained in this model is approximately 1,290,000.

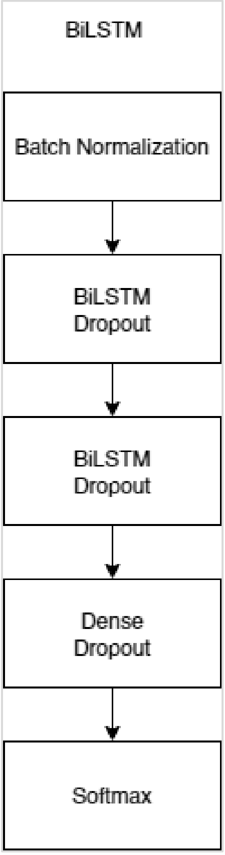

Figure 13: Model architecture using BiLSTM units

3.5.2. GRU

The architecture of the neural network using GRU units is similar to the BiLSTM network and is shown in Figure 14.

It has a normalization layer, two layers of GRU units with an additional Dropout layer to avoid overfitting the model, a Fully Connected layer, and a Softmax layer for the classification itself.

Since the GRU unit is internally simpler than the BiLSTM unit, the training process should be faster. In fact, this was the case in the experiments: training the model with GRU was an order of magnitude faster. This is a very relevant advantage in research.

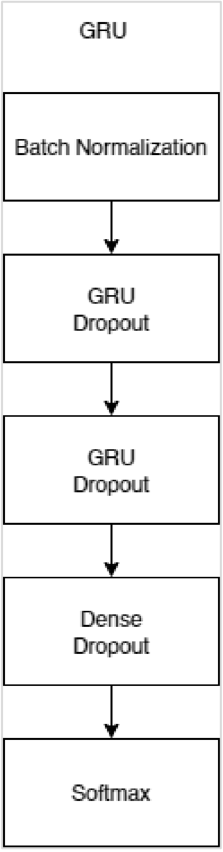

Figure 14: Model architecture using GRU

3.5.3. CNN

The final architecture of the network using convolutional layers is shown in Figure 15.

The model contains 6 convolutional blocks.

- Conv 2D (kernel size 5 by 5; ReLU activation function; padding=same)
- Batch Normalization
- Max Pooling 2D (size 4 by 4)
- Dropout (dropout value 0.2)

After the 6 convolution blocks, an intermediate Flatten layer is applied to obtain a one-dimensional structure by combining the parameters of all filters after the convolution operations. Then, identical to the previous architectures, a Fully Connected layer with 128 neurons and a Dropout and Softmax layer for classification.

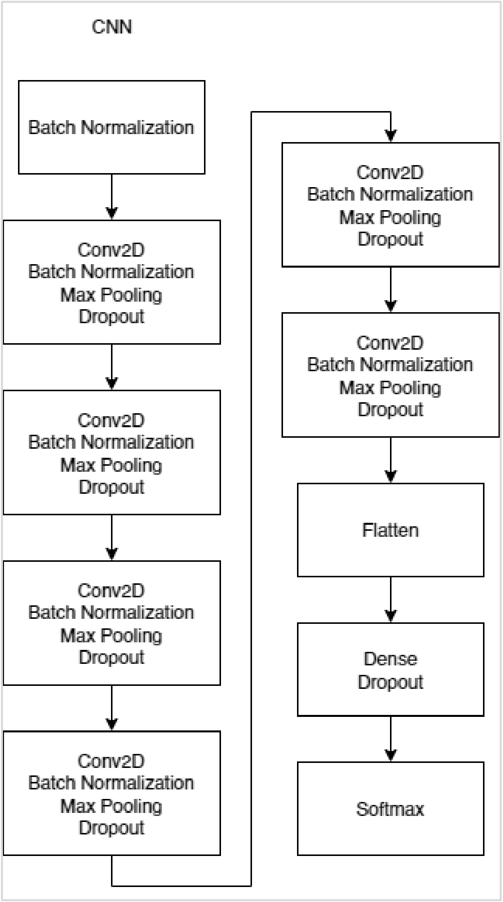

Figure 15: CNN model architecture

3.5.4. CRNN

The first part of the network consists of the same 6 convolutional blocks used in the regular CNN above. Then it is followed by an intermediate Reshape layer to transform the multidimensional data after convolution into the appropriate form for the recurrent layers.

The two recurrent layers have 256 and 512 LSTM units, respectively. They are followed by a Dropout layer with a value of 0.2, a Flatten layer to obtain a one-dimensional structure, a Fully Connected layer with 128 neurons, and a Dropout and Softmax layer for classification.

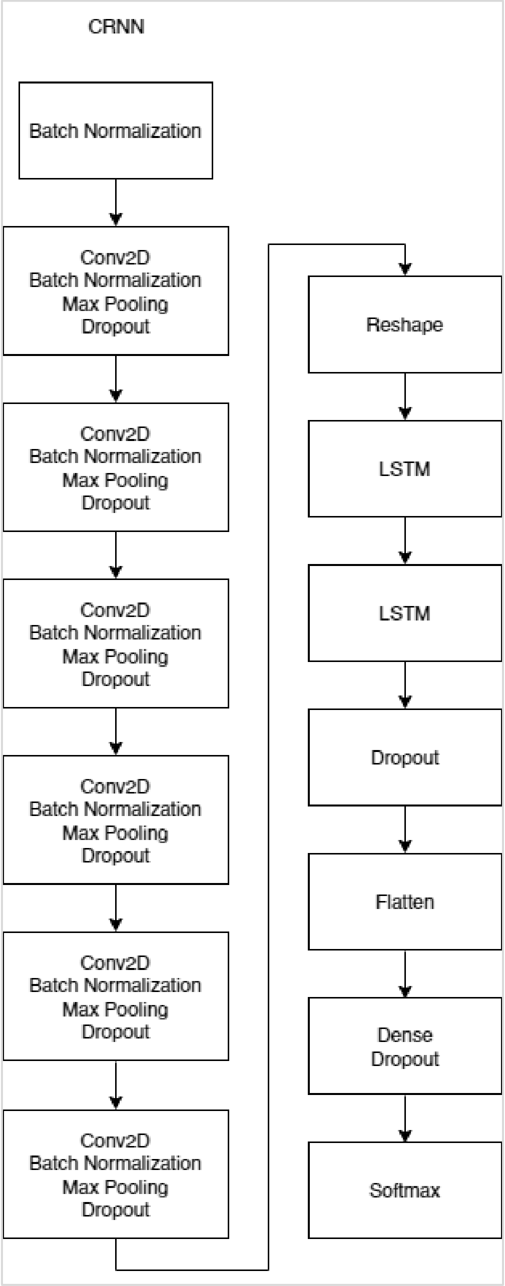

Figure 16: CRNN model architecture

The total number of parameters to be trained in this model is approximately 4 million.

3.6. Metrics for model evaluation

Evaluation metrics are key to the task of identifying the most efficient among a given set. To compare mathematical models of such a plan, there is a basic set of metrics that describe the model fairly objectively.

The metrics of accuracy, precision, recall and F1-score are commonly used to evaluate models. Let us look at each metric in more detail.

3.6.1. Accuracy

A standard metric for evaluating mathematical models that describes the closeness of measurement results to true values and is expressed as the ratio of the number of correctly classified data to the total number of data. It is calculated using the Formula 1.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \tag{1}$$

Of course, such a metric may not always be objective, especially if there is an imbalance of classes in the data set.

3.6.2. Precision

Also a standard model evaluation metric that describes the closeness of the measurements and is expressed as the ratio of correctly classified correct data to the total number of correct data. It is calculated using Formula 2.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{2}$$

Precision is especially important when it is necessary to reduce misclassification of true data (e.g., reducing the definition of absence of disease when disease is actually present). Always used with recall.

3.6.3. Recall

Also a standard metric that reflects the ability of the mathematical model to identify all correct data and is expressed as the ratio of correctly classified correct data to the total amount of data that was classified as correct. It is calculated using Formula 3.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{3}$$

Closely related to the previous metric, as a decrease in one metric increases the other, so a balance between the two should be achieved when developing the model.

3.6.4. F1-score

A metric used to evaluate the quality of classification, especially in cases where it is important to balance precision and recall of the model. It represents the harmonic mean between these two metrics and is intended for cases where the samples for classes are not balanced, or where precision and recall are equally important. It is calculated using Formula 4.

$$F_1 = \frac{2 * Precisin * Recall}{Precision + Recall} \quad (4)$$

The F1-score takes a value from 0 to 1, where 1 indicates an ideal model that has achieved maximum precision and recall (unfortunately, this is not currently possible). This metric is particularly useful in tasks where it is important to avoid both "false positives" (where a negative example is misidentified as positive) and "false negatives" (where a positive example is misidentified as negative).

All of these metrics have been used in previous researches and will be used in the experiments of this research.

3.7. Train and Test data splitting

During the experiments, the approach of splitting the data into training and test samples is used. For this purpose, there is a simple train-test split approach.

- The dataset is split into two parts: a training set and a test set
- The model is trained on the training set and then tested on the test set to evaluate the performance

However, this approach has some disadvantages, especially when the class distribution is not uniform. It is also less objective than the following method.

The most commonly used method in machine learning is K-Fold Cross-Validation. In this method, the dataset is divided into K subsets and the model is iteratively trained and evaluated K times. During each iteration, one of the subsets is used as a test set, while the other K-1 subsets are used to train the model. At each iteration, the metrics described in the previous section are recorded. After all iterations, the metrics are averaged to provide a robust evaluation of model performance.

Key Benefits of K-Fold Cross-Validation are the following.

- K-Fold Cross-Validation provides a more robust and less biased assessment of model performance than a one-time split into training and test sets
- Ensures that each data point is used exactly once for testing and that the model encounters different subsets of the data during training
- Helps evaluate how well the model generalizes to different subsets of the data, providing important insight into its robustness

The choice of K-value depends on the size of the dataset and the computing resources. Typically, a K-value from 5 to 10 is used. In general, K-Fold Cross-Validation is a valuable method for obtaining a more reliable evaluation of model performance, especially when working with limited amounts of data.

In fact, K-Fold Cross-Validation was used for the final evaluation of the models in the experiments of the research. The number of subsets for this validation was 5. However, for preliminary development and testing, train-test split was used for faster approximate results of model efficiency.

3.8. Development tools

The main development tool for experimentation is the Python programming language. It is widely used and well suited for machine learning, data processing, and neural network training. It also has a large number of packages that are essential for research.

In particular, the librosa library [18] is the basis for working with audio, especially for feature extraction and augmentation. It provides almost all the necessary functions to prepare the data before sending it to the input of mathematical models.

For model building, the Keras library is used, which is an add-on to Tensorflow and allows easy creation of models of the desired configuration.

Auxiliary libraries were numpy for working with data matrices and matplotlib for graphing results.

Development was done at JupyterLab on a system with the following configuration.

- AMD Ryzen 3 3600X
- 32GB RAM
- NVIDIA RTX 3060 (12GB)

Tensorflow was configured to use the resources of the graphics card for training.

4. Experiments

Using the algorithm described previously, two datasets of CREMA-D and IEMOCAP sound recordings were converted into matrices of numerical values. Each matrix is 128 by 40 in size.

A total of 8 experiments (with and without augmentation for each dataset) were performed. K-Fold Cross Validation was used to retrieve the metrics in each experiment, where K=5.

All trainings lasted 150 epochs and the batch size was 32.

The training graphs for both datasets are almost identical and do not carry much significance for comparison. Therefore, only the graphs for CREMA-D dataset will be presented.

First, let us examine the training results on CREMA-D and IEMOCAP datasets without data augmentation.

Figure 17 shows the model training plots using BiLSTM for CREMA-D dataset.

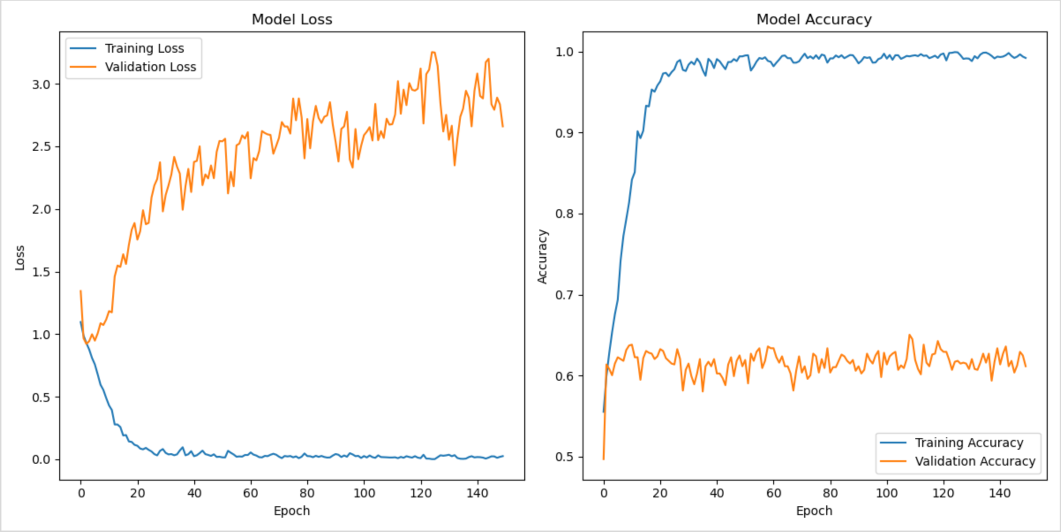

Figure 17: Training plots for BiLSTM model on CREMA-D dataset w/o augmentation

Figure 18 below shows the model training plots using GRU for CREMA-D dataset. We can see about the same trend as for BiLSTM model above.

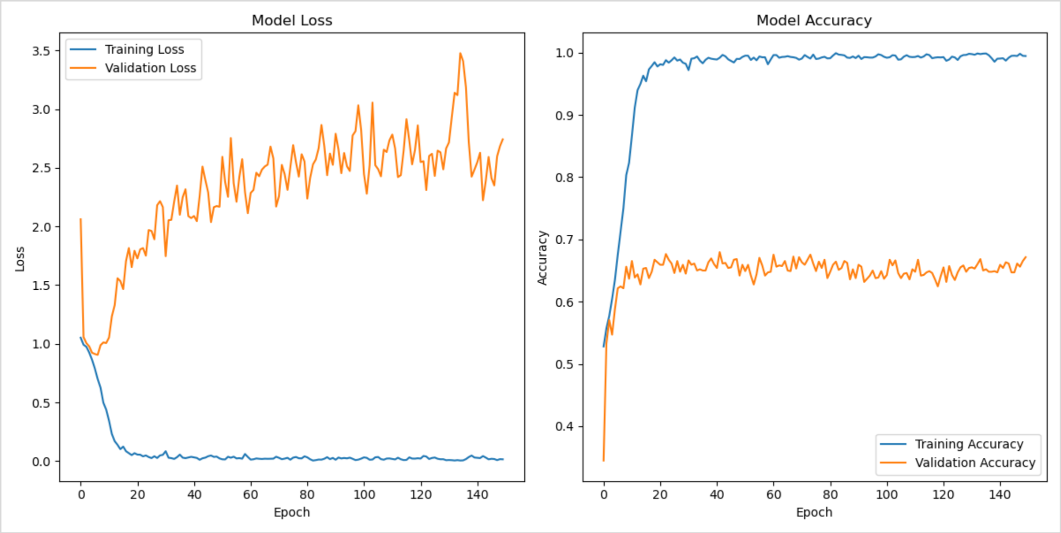

Figure 18: Training plots for GRU model on CREMA-D dataset w/o augmentation

Figure 19 shows the model training plots using CNN for CREMA-D dataset.

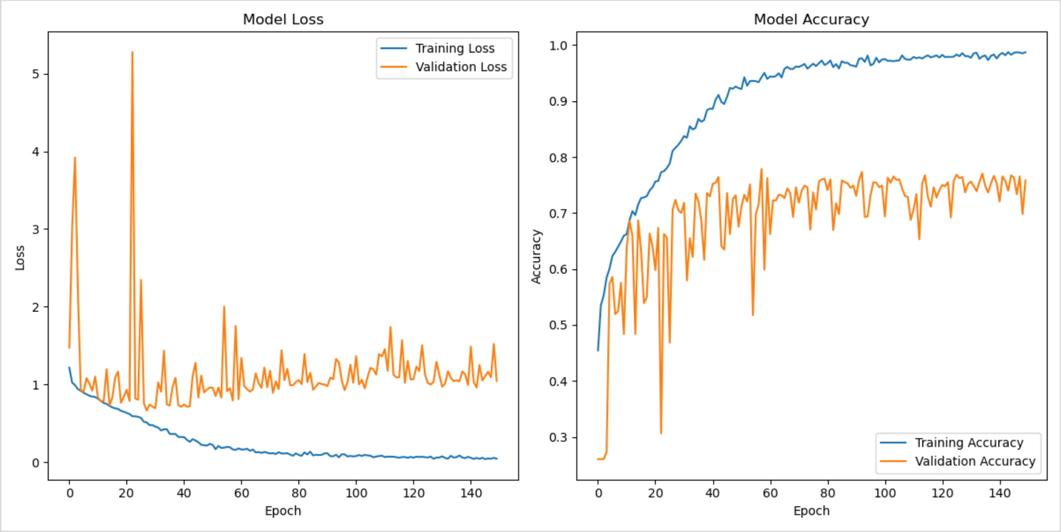

Figure 19: Training plots for CNN model on CREMA-D dataset w/o augmentation

Figure 20 shows the model training plots using CRNN for CREMA-D dataset. We also can find same trend as for the CNN model. So, recurrent and convolutional based models have about same trend in pairs.

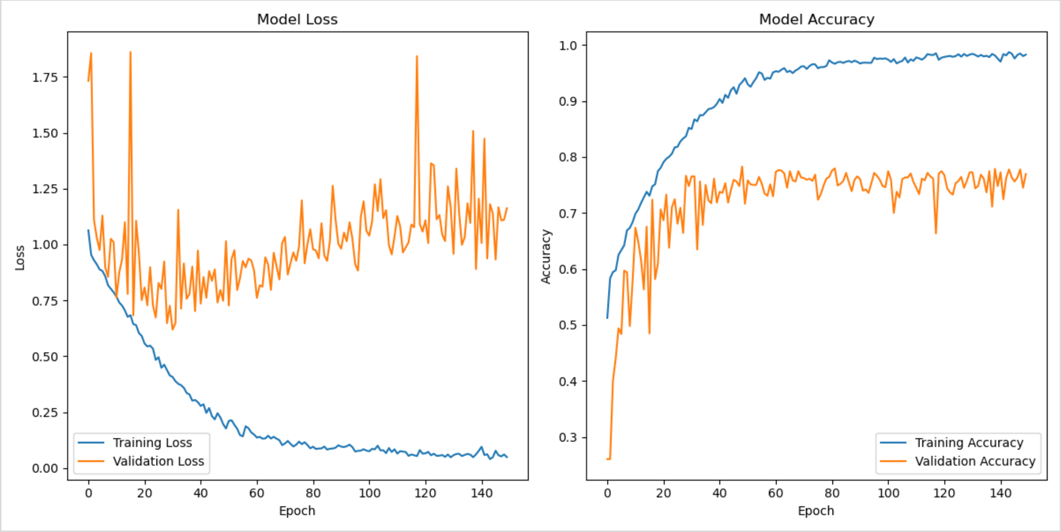

Figure 20: Training plots for CRNN model on CREMA-D dataset w/o augmentation

The results of the metrics for these experiments for both CREMA-D and IEMOCAP datasets are presented in Tables 2, 3, 4, and 5 (used “C” for CREMA-D dataset and “I” for IEMOCAP).

Table 2
Metrics of BiLSTM model on CREMA-D and IEMOCAP datasets w/o augmentation

	precision		recall		f1-score		support	
	C	I	C	I	C	I	C	I
anger	0.749	0.701	0.802	0.725	0.772	0.712	254	221
happiness	0.686	0.346	0.592	0.227	0.633	0.272	254	119
neutral	0.62	0.637	0.596	0.655	0.607	0.645	217	342
sadness	0.7	0.632	0.762	0.693	0.729	0.661	254	217
	0.691	0.613	0.691	0.624	0.688	0.616		
accuracy							**0.691**	0.624

Table 3
Metrics of GRU model on CREMA-D and IEMOCAP datasets w/o augmentation

	precision		recall		f1-score		support	
	C	I	C	I	C	I	C	I
anger	0.766	0.663	0.752	0.693	0.757	0.674	254	221
happiness	0.621	0.275	0.603	0.18	0.608	0.215	254	119
neutral	0.48	0.608	0.522	0.625	0.498	0.613	217	342
sadness	0.655	0.606	0.625	0.65	0.638	0.627	254	217
	0.636	0.577	0.63	0.589	0.63	0.579		
accuracy							**0.63**	0.589

Table 4
Metrics of CNN model on CREMA-D and IEMOCAP datasets w/o augmentation

	precision		recall		f1-score		support	
	C	I	C	I	C	I	C	I
anger	0.828	0.732	0.789	0.754	0.803	0.742	254	221
happiness	0.727	0.414	0.71	0.19	0.718	0.259	254	119
neutral	0.678	0.632	0.732	0.714	0.698	0.669	217	342
sadness	0.799	0.686	0.762	0.704	0.771	0.684	254	217
	0.761	0.641	0.749	0.652	0.749	0.636		
accuracy							**0.749**	0.652

Table 5
Metrics of CRNN model on CREMA-D and IEMOCAP datasets w/o augmentation

	precision		recall		f1-score		support	
	C	I	C	I	C	I	C	I
anger	0.78	0.781	0.826	0.704	0.79	0.733	254	221
happiness	0.724	0.402	0.666	0.276	0.689	0.317	254	119
neutral	0.668	0.636	0.79	0.69	0.72	0.659	217	342
sadness	0.856	0.66	0.691	0.713	0.758	0.681	254	217
	0.76	0.646	0.741	0.644	0.74	0.637		
accuracy							**0.741**	0.644

Among these results, we can see that BiLSTM performs better in the category of recurrent networks. The CNN performs similarly. Let us examine the results of these models when augmentation is applied to the data.

Figures 21 and 22 show the training plots of BiLSTM and CNN on CREMA-D dataset using augmentation.

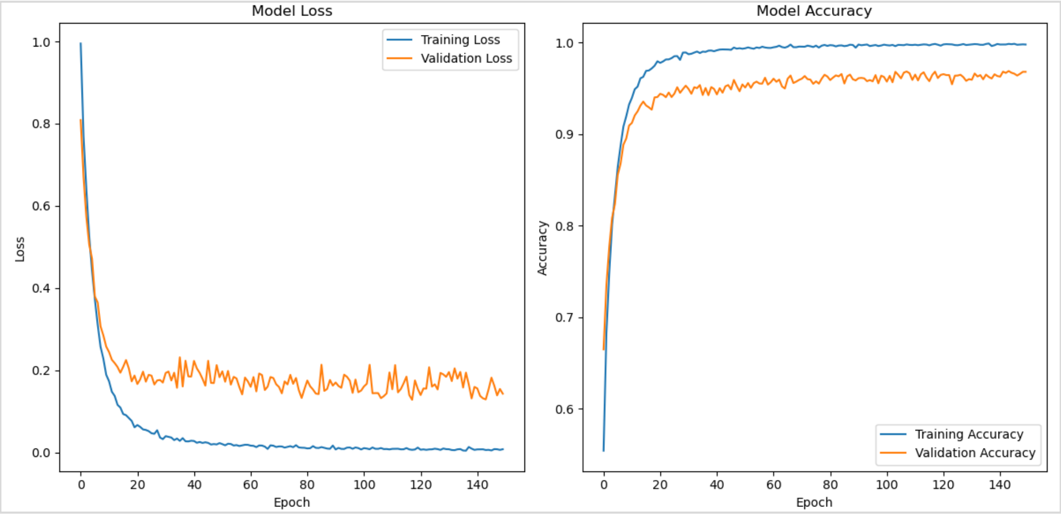

Figure 21: Training plots for BiLSTM model on CREMA-D dataset with augmentation

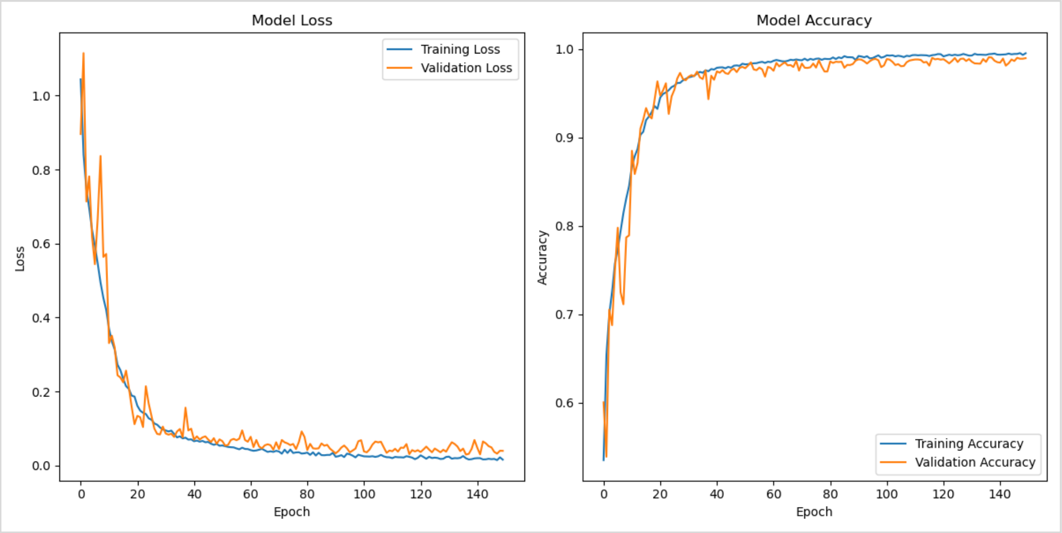

Figure 22: Training plots for CNN model on CREMA-D dataset with augmentation

The results of the metrics for both datasets for BiLSTM and CNN are presented in Tables 6 and 7.

Table 6
Metrics of BiLSTM model on CREMA-D and IEMOCAP datasets with augmentation

	precision		recall		f1-score		support	
	C	I	C	I	C	I	C	I
anger	0.978	0.969	0.979	0.962	0.979	0.965	1 525	1 324
happiness	0.969	0.945	0.96	0.884	0.965	0.913	1 525	714
neutral	0.944	0.936	0.963	0.947	0.953	0.942	1 304	2 050
sadness	0.969	0.927	0.958	0.95	0.963	0.938	1 525	1 301
	0.966	0.943	0.965	0.943	0.965	0.943		
accuracy							**0.965**	0.943

Table 7
Metrics of CNN model on CREMA-D and IEMOCAP datasets with augmentation

	precision		recall		f1-score		support	
	C	I	C	I	C	I	C	I
anger	0.998	0.993	0.992	0.992	0.995	0.993	1 525	1 324
happiness	0.992	0.993	0.987	0.953	0.989	0.973	1 525	714
neutral	0.977	0.981	0.986	0.983	0.981	0.982	1 304	2 050
sadness	0.985	0.967	0.987	0.987	0.986	0.977	1 525	1 301
	0.988	0.982	0.988	0.982	0.988	0.982		
accuracy							**0.988**	0.982

5. Results

So, let us analyze the obtained results.

In Figure 23, we can see the accuracy of each model on both datasets without and with augmentation.

The developed GRU model performed the worst on a small amount of data, while the CNN model performed the best on both CREMA-D and IEMOCAP.

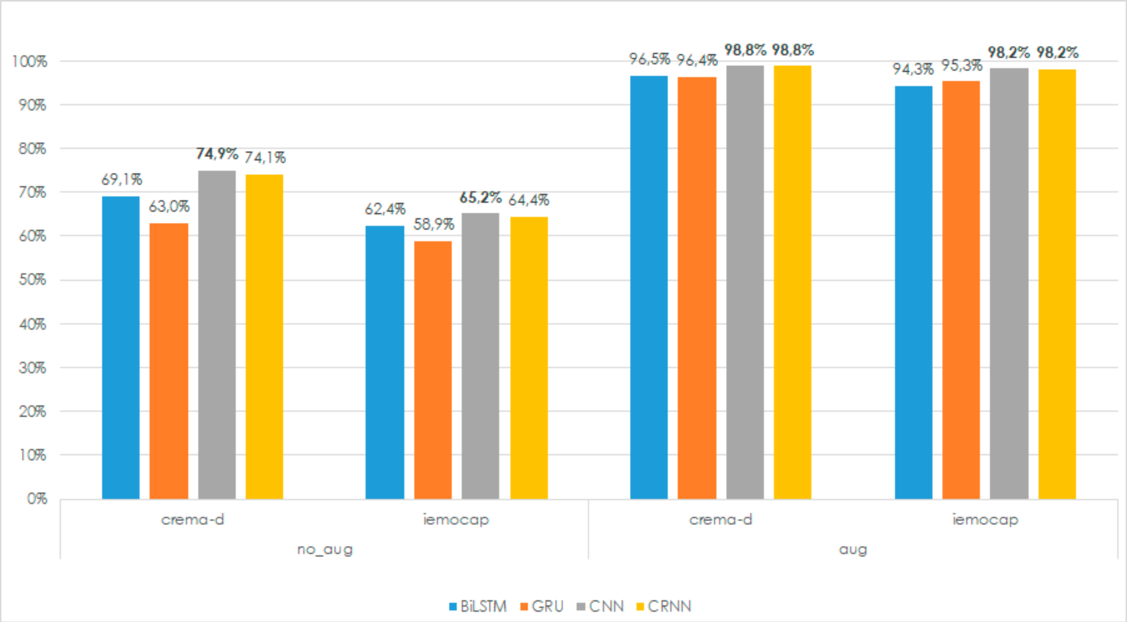

Figure 23: Results comparison

It can be seen that the non-uniformity of the classes in the IEMOCAP dataset has a somewhat negative effect on the quality of the models, especially when the size of such a dataset is relatively small. For all models on the IEMOCAP dataset, the accuracy is on average 7% lower.

Looking at the training plots, we can also conclude that recurrent networks are trained much faster than convolutional networks on such data, but the loss function graph starts to grow after a certain point, indicating overfitting. Convolutional networks also show unstable loss function trend, that indicates less of data to learn the patterns.

Nevertheless, the convolutional model performed better on both datasets, and this trend continued in the experiments with augmented data.

With augmented datasets, all models showed approximately the same accuracy. However, both BiLSTM and GRU show poor results compared to the convolutional layer models. In addition, the likely influence of class non-uniformity is balanced, and GRU shows even higher accuracy than BiLSTM (in the case of IEMOCAP dataset).

The percentage improvement statistics are shown in Figures 24 and 25.

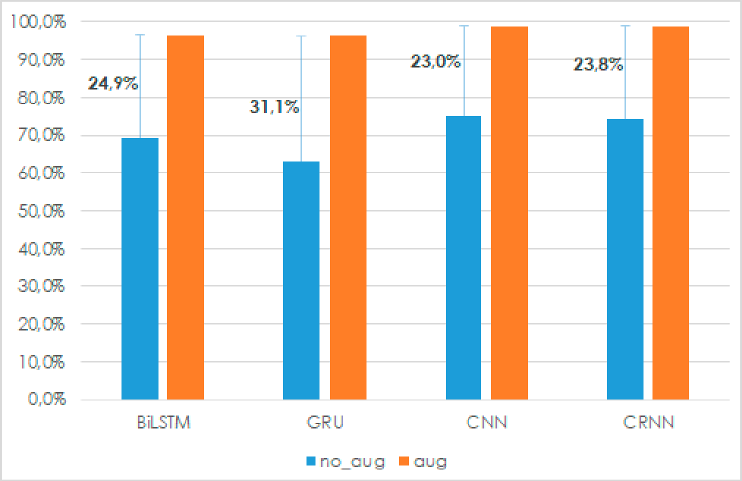

Figure 24: Increase in accuracy for CREMA-D dataset

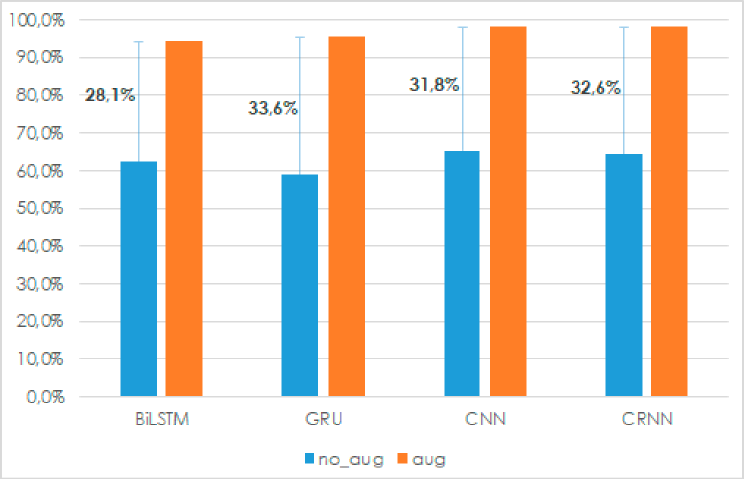

Figure 25: Increase in accuracy for IEMOCAP dataset

The other models also improved in accuracy: while without augmentation the results on the IEMOCAP dataset were about 7% lower, with augmentation the gap was reduced to 1-2%, indicating the importance of augmentation and the number of data samples.

The figures show that augmentation boost on IEMOCAP dataset was 31-32% compared to 26-27% for CREMA-D.

If we look at the f1-score for the two datasets without augmentation, we can clearly see the impact of class heterogeneity. For example, for the BiLSTM model on IEMOCAP dataset, the f1-score for “happiness” is only 0.27, while for the other classes this value is above 0.6. On the other hand, CREMA-D dataset has no such problem.

Looking at the same model, but already on the extended data, it can be seen that the difference of the f1-score value between the classes has significantly decreased and is for all of them above 0.9. Although it can still be seen that the lack of instances of a class in dataset still affects the recognition accuracy of that class.

6. Discussions

The results obtained are in line with those obtained by previous researchers in this domain. As in the other researches, the CNN approach was found to be the most effective among the researches considered.

It can also be seen that.

- Augmentation plays a very important role in SER for small datasets
- MFCCs as audio features are sufficient to obtain sufficiently high quality models

Moreover, the results obtained are also better than all the reviewed works. In this paper, a dynamic frame size approach was used, which was not mentioned in other works. Nevertheless, this approach has shown a successful impact on the results. In the pre-development period, before the dynamic frame size approach was used, the results were much worse.

7. Conclusions

In this research, we presented the results of experiments on emotion recognition from speech using CREMA-D and IEMOCAP datasets, combined with the extension of the datasets through augmentation.

We developed four models: BiLSTM, GRU, CNN and CRNN. The result shows the following.

- Among the investigated models, the convolutional layer model (CNN) shows the best quality result
- CREMA-D dataset shows on average slightly higher accuracy results in all the experiments conducted

Data augmentation also played a significant role, improving the accuracy of all models by an average of 25-30%.

The results of this research show the superiority of the described approach over the analyzed works of its predecessors.

The more data and evenly distributed classes in a dataset, the more effective the model can be developed. Despite increased resource use, a large number of samples is advantageous. While augmentation improves model quality, its efficiency on real data may only be approximate.

The described methods can be improved and explored in other conditions, extending research to include visual information in SER and examining models with more than four classes. Even now, the results can inform software development for various applications.

Acknowledgements

We would like to express our deep gratitude to the research advisor who, in this difficult time, was able to give us the maximum amount of time necessary for the quality of the research. She shared her knowledge and experience and pushed our work in the right direction.

We thank all the co-authors who participated and helped in the research and preparation of the paper. We thank the University of Southern California for preparing IEMOCAP dataset and allowing us to use it in this research.

And of course, our heartfelt thanks to our families for their support, inspiration, belief in our abilities, for teaching us perseverance, the ability to achieve goals, and for helping us find the right path.

References

[1] Emotional speech recognition using deep neural networks. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8877219/.

[2] Speech emotion recognition using deep learning. URL: https://blog.dataiku.com/speech-emotion-recognition-deep-learning.

[3] Speech Emotion Recognition with deep learning. URL: https://www.sciencedirect.com/science/article/pii/S1877050920318512.

[4] Speech emotion recognition using deep learning techniques: A review. URL: https://www.researchgate.net/publication/335360469_Speech_Emotion_Recognition_Using_Deep_Learning_Techniques_A_Review.

[5] A deep learning approach for speech emotion recognition optimization using meta-learning. URL: https://www.mdpi.com/2079-9292/12/23/4859.

[6] P. Srinidhi, Speech based emotion recognition, Int. J. Res. Appl. Sci. Eng. Technol. 10.6 (2022) 3160–3165. doi:10.22214/ijraset.2022.44583.

[7] Surrey audio-visual expressed emotion (SAVEE) database. URL: http://kahlan.eps.surrey.ac.uk/savee/.

[8] L. S. R, R. F. A, The ryerson audio-visual database of emotional speech and song (RAVDESS), 2018. URL: https://zenodo.org/records/1188976.

[9] Toronto emotional speech set (TESS) | TSpace Repository. URL: https://tspace.library.utoronto.ca/handle/1807/24487.

[10] GitHub - cheyneycomputerscience/crema-d: crowd sourced emotional multimodal actors dataset (CREMA-D). URL: https://github.com/CheyneyComputerScience/CREMA-D?tab=readme-ov-file.

[11] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, S. S. Narayanan, IEMOCAP: interactive emotional dyadic motion capture database, Lang. Resour. Evaluation 42.4 (2008) 335–359. doi:10.1007/s10579-008-9076-6.

[12] Bevor Sie zu YouTube weitergehen. URL: https://www.youtube.com/@ValerioVelardoTheSoundofAI.

[13] How to train MFCC using machine learning algorithms. URL: https://www.tutorialspoint.com/how-to-train-mfcc-using-machine-learning-algorithms.

[14] What are recurrent neural networks? | IBM. URL: https://www.ibm.com/topics/recurrent-neural-networks.

[15] StatQuest with Josh Starmer, Long short-term memory (LSTM), clearly explained, Відео, 2022. URL: https://www.youtube.com/watch?v=YCzL96nL7j0.

[16] S. Kostadinov, Understanding GRU networks, 2017. URL: https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be.

[17] Krish Naik, Bidirectional RNN indepth intuition- deep learning tutorial, Відео, 2020. URL: https://www.youtube.com/watch?v=D-a6dwXzJ6s.

[18] Librosa — librosa 0.10.1 documentation. URL: https://librosa.org/doc/latest/index.html.