Evaluation of a Sliding Window mechanism as Data Augmentation in Emotion Detection on Speech

Matheus Almeida Farias da Silva[1], Tiago da Silva Almeida[1] and Rafael Lima de Carvalho[1]

[1] *Federal University of Tocantins, Computer Science Department, Tocantins, Brazil*

Reception date of the manuscript: 12/02/2021
Acceptance date of the manuscript: 01/04/2021
Publication date: 13/04/2021

Abstract— Emotion analysis is an important field of study, with many applications for security, financial, and politician. Despite being a subjective branch of study, emotion analysis can be simulated by Machine Learning algorithms that are trained for this purpose, through cataloged audio datasets, they can recognize patterns in these media that could be related to corresponding emotion. Neural Network Algorithms are able to work on the recognition of these emotions, with a focus only on audio, known as Speech Emotion Recognition (SER). Neural Network Algorithms generally obtain unequal averages of referring results such as recognition of emotions when applied to different audio datasets. This research evaluates a Data Augmentation method called Slide Window, which generates more data samples in order to increase the averages of classification rates. The method has been applied to three public datasets: EMO-DB, SAVEE, and RAVEDESS. The experiments have shown effectiveness in the increasing of the recognition rates of about to 11.95% on the EMO-DB base, 22.76% on SAVEE, and 18.82% on RAVEDESS when compared to other approaches in the literature.

Keywords— Speech Emotion Recognition, Voice processing, Machine Learning, Deep Neural Networks

I. Introduction

Speech Emotion Recognition (SER) is the process of inferring the emotion through machine learning using speech as data input [1]. Thus, this field includes algorithms to extract spectral features related to voice inputs [2]. In this way, the further step consists of the machine automation in order to find a model capable of identifying patterns using those features.

Towards investigating the SER problem, Jiang *et. al.* [1] proposed an algorithm called Parallelized Convolutional Recurrent Neural Network, (PCRNN), which is composed of two modules: A) CNN (Convolutional Neural Network) and B) LSTM (Long Short-Term Memory). Each audio sample is submitted to both modules, and the final scores are ensembled in order to produce a unique output. Module A is responsible to learn details of emotion using features such as speech frequency. On the other hand, module B tries to identify temporal changes in emotions. Their proposed work was submitted to four labeled audio dataset, known as EMO-DB, CASIA, ABC, and SAVEE. For which, they have calculated Weighted Average and Average on the data of confusion matrix.

In [3] an emotion recognition algorithm is developed with a 2D CNN architecture in which it intends to optimize training time and enhance the selection of key data, with the main focus on audio pre-processing. It means to focus its process on eliminating noises and intervals without any speech. Their experiments were carried out on the IEMOCAP and RAVDESS datasets. Furthermore, it is mentioned that the focus of execution in the pre-processing was centralized by the fact that the current CNN architectures did not reveal any significant improvement in terms of precision and cost complexity in the processing of voice signals, and the use of LSTM is useful for training sequential data, but they are difficult to train effectively and are more computationally complex.

In the pre-processing stage of [3], the audio files undergo noise filtering and removal of irrelevant information using a function that takes the energy and amplitude data as input, producing a single value for the analyzed audio segment. After this stage, the spectrogram of these audios is used as a two-dimensional input representing the strength of the audio signal by the different frequencies.

Zhao *et. al.* [4] proposed two architectural models (1D CNN LSTM and 2D CNN LSTM) for recognition of emotions using voice. In order to increase the accuracy of emotion recognition, the authors proposed a learning block called Local Feature Learning Block (LFLB), which is responsible for the convolutional learning process. This block consists of a convolutional layer, a normalization layer, an exponential linear unit (ELU) layer, and a Max-Pooling layer, which is used to reduce the scales of the characteristic matrix. The algorithm contains three LFLBs, which generate a partial result to be inserted into an LSTM network (final classifier).

Contact data: Rafael Lima de Carvalho, rafael.lima@uft.edu.br

The 1D model uses the raw audio data as input, while the 2D uses the log of Mel scale as its input. The algorithm has been submitted under the German EMO-DB and the American IEMOCAP datasets.

Therefore, this work aimed to build a Neural Network based on [4] and evaluate the effectiveness of a Data Augmentation method called the Sliding Window under the EMO-DB, RAVDESS, and SAVEE datasets. The considered Sliding Window method consists of creating new audio files from the original files of each base. A fixed size window is applied to the audio, generating new audio files with the same size.

The paper is structured as follows. In Section II we describe the Mel-Frequency Cepstral Coefficients and the Deep Neural Network algorithms used in the experiments. In Section III we describe the Sliding Window mechanism for Data Augmentation as well as the benchmark datasets. In addition, in Section IV we describe the experiments and the results about the evaluated methodology. Finally, Section V brings the final remarks of the proposed work.

II. BACKGROUND

In this section we present the features and classifiers employed in this research.

a. Mel-Frequency Cepstral Coefficients

Psychophysical studies have shown that human perception of speech sound frequency does not follow a linear scale [5]. The Mel scale is a method that tries to reproduce that perception. It collects speech parameters similar to those used by humans to hear speech while considering all other information [6]. Thus, for each real frequency sonorous tone, f, measured in Hz, a concurrent tone is measured on a scale called the Mel scale f_{mel},

$$f_{mel} = 2595 \log_{10}\left(t1 + \frac{f}{700}\right), \tag{1}$$

which is a scale that aims to imitate the unique characteristics perceived by the human ear.

The normalization of frequencies in Mel scale consists of two main steps: i) the audio frames (small audio segment) are filtered in a process called Fast Fourier Transform (FFT), for which filters are applied with the average of the spectrum around the central frequency, and have different window sizes. These window sizes refer to the number of samples (of raw audio). In this way, the larger the window, the fewer filters are generated. This transformation process is called filter bank; ii) the resulting frequencies from step i) are normalized to the Mel scale [7].

In order to generate Mel-Frequency Cepstral Coefficients (MFCC), the first step is to divide the audio signal into frames, after that, they will be applied to the FFT. Second, filter bank processing is performed in the power spectrum, using the Mel scale. Obtaining the MFCC coefficients can be described as,

$$\hat{C}_n = \sum_{n=1}^{k} \left(\log \hat{S}_k\right) cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{k}\right] \tag{2}$$

where k is the number of Mel coefficients, $\hat{S}_k$ is the output of the filter bank, and $\hat{C}_n$ is the final MFCC coefficient.

b. Classifiers

1. Long Short-Term Memory

LSTM contains special units called memory blocks in the recurring hidden layer. The memory blocks contain memory cells with self-connections that store the temporal state of the network, in addition to special multiplicative units called gates to control the flow of information and assign of priorities [8].

2. Convolutional neural network

CNN has been used for pattern recognition tasks, such as facial recognition and handwritten numeric recognition [9]. CNN learns to map a particular image, in this case, a matrix of values, to its corresponding category, detecting a series of specific characteristics of each entry [10]. A CNN can have dozens or hundreds of layers, where each one learns to detect different characteristics of a matrix.

III. METHODOLOGY

This section presents the construction techniques and methods that were the basis for obtaining the results of the study proposed in this work. In subsection a it is shown the datasets used in the experiments, with basic information such as the amount of samples for each class (emotion). In addition, in subsection 2 we explain how the sliding window mechanism works. In the further subsections we describe the evaluation functions as well as the employed classifier.

a. Datasets

1. Surrey Audio-Visual Expressed Emotion

Surrey Audio-Visual Expressed Emotion (SAVEE) is an audiovisual dataset that was designed with the purpose of being applied in the development of automatic emotion recognition systems [11]. It consists of recordings from 4 male actors, graduate students, and researchers from the University of Surrey aged between 27 and 31, with a total of 7 emotions. Emotions are described psychologically in discrete categories: anger, disgust, fear, happiness, sadness, and surprise. Neutral emotion was also added to provide the 7 desired categories.

2. Ryerson Audio-Visual Database of Emotional Speech and Song

The Ryerson Audio-Visual Database of Emotional Speech and Song, called RAVDESS, is a Canadian multimodal dataset of speech and music with a focus on emotional analysis that has been entirely recorded in English. RAVDESS consists of 24 professional actors, each performing 104 unique vocalizations with emotions that include: happiness, sadness, anger, fear, surprise, disgust, calm and neutral [12]. Because it is multimodal, it is divided into three modules. In the first, audio-visual (AV), each file contains both the video and audio recording. The second, video-only (VO), consists

only of the video of actors facial expressions. And finally, the audio-only (AO), containing only the recorded audio.

3. Berlin Database of Emotional Speech

Containing speeches of 10 actors, 5 women and 5 men, the Berlin Database of Emotional Speech (EMO-DB) [13] is a German dataset of audios for the analysis of emotion. The actors generated 10 phrases each, half of which were long phrases, the other half, short phrases and some extra phrases, which can be used in daily communication and are interpretable in all applied emotions. Emotions are divided into 7 categories: neutral, anger, fear, joy, sadness, disgust, and boredom. The recordings were made in an anechoic chamber with high-quality recording equipment. The material contained in the dataset comprises about 800 phrases.

b. Data and Pre-Processing

1. MFCC

The library Librosa [14] has methods of extracting the audio characteristics, among them the MFCC, being able to choose the number of coefficients to be generated. Whereas, the greater the number of coefficients, the more detailed the data returned will be. Thus, in this work, 128 coefficients were used, with the FFT with a window size of 2048 frames per filter, and a shift of 512 frames per filter. In order to keep all audios at the same size, shorter audios have been extended up to 8 seconds, by concatenating their own content. On the other hand, audios greater than 8 seconds have been truncated to this standard size. Consequently, as all audios have 8 seconds, the resulting matrix will be 128x251 for all audios, with the first value being the number of coefficients and the second the number of resulting values.

2. Sliding Window Mechanism

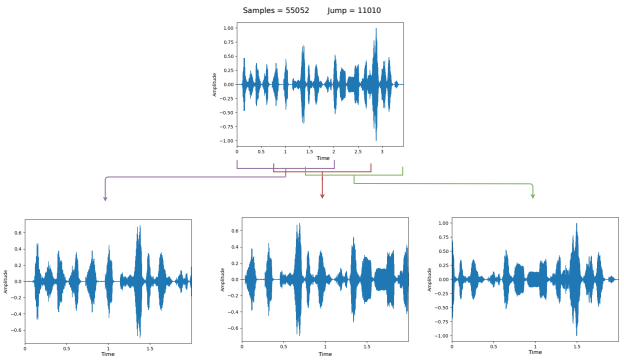
Overfitting is a very recurring problem found in neural network training [15]. It makes the machine learning algorithm very good at recognizing the patterns of the training data but does not get as good at recognizing the test data. Overfitting limits the generalization of the algorithm. Some techniques used during the construction of the neural network algorithm can avoid this problem, one of which is Data Augmentation, which is related to the generation of new samples from the existing data, in order to increase the size of the dataset [16]. Thus, the neural network algorithm receives a larger amount of data, both for training and for testing, and consequently, its generalization capacity will be greater, since it will be able to recognize more data samples.

For this purpose, this work used a method called Sliding Window, as a way to perform the Data Augmentation technique. Once the window and shift sizes are defined, the window is positioned at the beginning of the audio data, cropping the file as it slides over it. Each cropped area is a new sample of the same class.

Window Structure

In order to process the audios, a sample rate is defined. The sample rate represents the number of frequency samples to be collected per second. This work uses a sample rate of 16.000 (sixteen thousand), thus, with a Sliding Window of 2 seconds, each sample is composed of 32.000 frequencies. In addition, audio files with less than 2.5 seconds are not suitable for running the sliding window mechanism, so they have been avoided by this procedure. An example where a single audio with 3.44 seconds became three new ones with 2 seconds each is shown in Fig. 1.

Fig. 1: Audio multiplication in three new audios through the Sliding Window process.



Slide Window Shift Size

As we can see in Section III, the audio files have different sizes. If we establish a fixed number for the shift of the window, this would cause an extremely unbalanced augmented dataset. In order to overcome this problem, the shift size is calculated using the equation 3.

$$S = \frac{tam}{Df} \tag{3}$$

where S is the amount of frequency data that the Sliding Window moves and tam the total size of frequency data for the audio input. The Df (coefficient of division) determines the size of the Sliding Window shift by dividing the number of frequency samples of the original audio. In this work, we defined $Df = 5$ as a good value based on the fact that the minimum accepted audio length (2.5 seconds) in the Sliding Window process and the size of the resulting audios (2 seconds). Establishing $Df = 5$ guarantees that at least two audio files (with 2 seconds each) will be generated for audio files with the minimum acceptable size.

Dataset transformation

The amount of audios generated from the Sliding Window process is directly related to the duration of each audio input, so the growth on the number of samples may vary for different emotions and datasets. If we take EMO-DB dataset we find audio samples that vary is size from 1.12 to 8+ seconds. In Table 1 it is shown the distribution of audio files by intervals of seconds. It can be seen in Table 1 that the RAVDESS dataset has practically all of its data over the 3 to 4 seconds category.

The augmentation data generated by the sliding window mechanism for each dataset and each emotion is shown in Table 2. We observe that the growing is different for each dataset. This occurs due to the duration of the audio samples. Samples with greater audio sizes produces more new

TABLE 1: DISTRIBUTION OF THE AUDIO DURATION OVER THE EMOTION DATASETS.

Audio length category	EMO-DB	RAVDESS	SAVEE
1 to 2 seconds	126	0	13
2 to 3 seconds	224	1	92
3 to 4 seconds	136	997	191
Greater than 4 seconds	49	250	184
TOTAL	**535**	**1248**	**480**

samples. Looking at the datasets RAVDESS and SAVEE, we notice that their audio samples have an average length of 3.42 and 3.49 seconds. This implied a growth that remained above 200% . In contrast, the EMO-DB, with 2.46 seconds of average, obtained a lower growth, for which only half of its audios had been applied to the Sliding Window process.

TABLE 2: DATA GROWTH AFTER THE SLIDING WINDOW PROCESS.

EMO-DB			
Emotion	**Original**	**With Window**	**Growth**
Anger	127	287	126%
Boredom	81	195	141%
Disgust	46	141	207%
Fear	69	119	72%
Happiness	71	150	111%
Neutral	79	143	81%
Sadness	62	223	260%
TOTAL	**535**	**1258**	**135%**

RAVDESS			
Emotion	**Original**	**With Window**	**Growth**
Neutral	96	364	279%
Calm	192	759	295%
Happiness	192	748	290%
Sadness	192	755	293%
Anger	192	762	297%
Fear	192	728	279%
Disgust	192	764	298%
TOTAL	**1248**	**4880**	**291%**

SAVEE			
Emotion	**Original**	**With window**	**Growth**
Anger	60	209	248%
Disgust	60	224	273%
Fear	60	211	252%
Happiness	60	221	268%
Neutral	120	422	252%
Sadness	60	245	308%
Surprise	60	218	263%
TOTAL	**480**	**1750**	**264.5%**

c. Training methods

In this work, we consider the Speaker-Dependent (SD) approach to considering the training of the machine learning algorithm. In this method, the dataset is split into two parts: training and test. When separating the data, the algorithm is trained using the training data set and the evaluation is done using the test set. In the context of Voice Recognition, the name Speaker-Dependent means that the algorithm depends on the data of all Actors contained in the dataset, both in the test stage and in the training. In this work, the use of the Speaker-Dependent method considering 80% of data for training and the remaining 20% for tests, with stratification.

1. Evaluation functions

Unweighted Average

The Unweighted Average (UA) is an arithmetic average based only on the true positives, disregarding the other possible results obtained by the classes. This way of evaluating the algorithm performance with the Confusion Matrix is presented in works related to SER, as in [4] and in [1]. This work uses this evaluation method as the main reference, because it is one of the most used, in addition to being a great metric when the data are balanced.

ROC-AUC

ROC-AUC is also another way to assess the sensitivity of the algorithm in classifying test data. ROC (Receiver Operating Characteristic) is represented by a graphical curve generated from a function that calculates the true positives with the false positives. In this case, a unique number (score) named AUC (Area Under the Curve) shows the area that the curve copper is used to evaluate the algorithm. The higher the ROC-AUC, the greater the ability of the algorithm to distinguish between classes.

F1-Score

Divided into Macro and Micro, the F1-Score is an evaluation metric on the test data that aims to show the degree of precision the algorithm has, taking into account the balance of each class. In this sense, the more similar the algorithm precision (in the case of this work, the UA), the more the data is balanced, proving its precision veracity. For the F1-Score with the Micro approach, the F1-Score measures a single value taking into account the balance of the classes as a whole, whereas the Macro, originates from the calculation with each class.

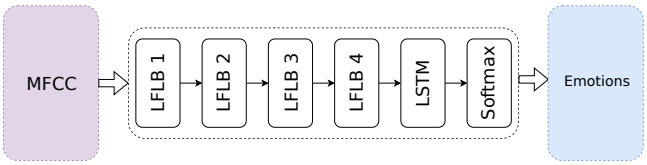
d. Classification Algorithm

Making a computer recognize an emotion requires, mostly, a deep and broad analysis in relation to the audio characteristics. It is important to specify how each value consists, as well as the relationship of these values as a whole over time. With this in mind, this work used the same algorithm model present in [4]. The model is composed of four Local Feature Learning Blocks (LFLB), which are blocks composed of layers with focused CNN 2D to abstract as much relevant information from the audio as possible, and an LSTM network layer. This scheme has been selected with the objective of analyzing dependencies between a sequence of data. Its main objective is to identify which are the dependencies of temporal context from the data resulting from the LFLB, and finally a Softmax layer to the generalization of logistic regression for classification problems with several classes. Those classes are the emotions of the audio according to the learned characteristics. In Fig. 2 it is shown the arrangement of the algorithm used for this work.

1. CNN 2D

In order to obtain a higher level of abstraction of which audio characteristics result in the classification of a certain emo-

Fig. 2: Architecture diagram of the algorithm for the classification of emotions from the MFCC extracted from the audios [4].



tion, the layers of the CNN 2D neural network were used for the composition of the LFLBs. These types of layers rely on the most outstanding data filtering, which is done through a kernel. For all convolution layers, a 3x3 size kernel is employed, that is, 9 units of the characteristics matrix passed through the filtering at a time, with a 1x1 Stride, which informs how many units the kernel has moved until its next filtering step. A small number of Stride is also used in order to capture more information about the audio input.

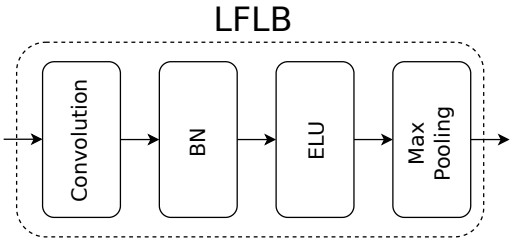
2. LSTM

The LSTM layer in this algorithm has the purpose of examining the composition and variation of the audio data coming from the CNN layers through a temporal analysis. It is responsible for generating the latest classification data of the neural network. Each cell (memory block) composing the LSTM layer performs an analysis calculation on the inserted data, thus, a total number of 256 LSTM cells is used, twice the output value of the last CNN layer. This is necessary to increase the reliability of the generated result.

3. Local Feature Learning Block

Each LFLB is composed of a Convolutional layer, a Batch Normalization (BN), an ELU layer (exponential linear unit), and a Max Pooling layer, with the Convolutional and Max Pooling layers being the fundamentals of this set [4]. Thus, the composition of an LFLB and the corresponding operation order are shown in Fig. 3.

Fig. 3: Composition of a Local Feature Learning Block [4].



As mentioned in Subsection 2, the Convolutional layers filter the input data as a way of mapping the striking features. These features are also called activations. The BN layer normalizes the results of the Convolutional layer of each batch of trained audios, this implies an improvement in performance and stability of the learning layers. The transformation applied by BN keeps the average activation close to 0 and the standard deviation of activation close to 1. The BN layer is then connected to the ELU that calculated the BN output. It takes the average of the activation values towards zero, in order to decrease the learning time and increase the level of recognition. Finally, the Max Pooling layer takes the most important features to stand out over the distortions and noise contained in the audios. It manages to make this abstraction by dividing the input into regions and giving the maximum values for each region as outputs.

4. Layer parameters

In order to make sure the correct execution of the classification algorithm, it is important to pay attention to some implementation details. One of these details is the control of data produced from each layer. The size of the output of each LFLB is defined by the kernel size, i.e. the region that filters the activations of the layer, and the Stride that tells how many units the kernel must move after activation. And for the LSTM layer, the size of its output is defined by how many units of LSTM the layer is composed of. Table 3 presents the control configuration used in this work.

IV. Results

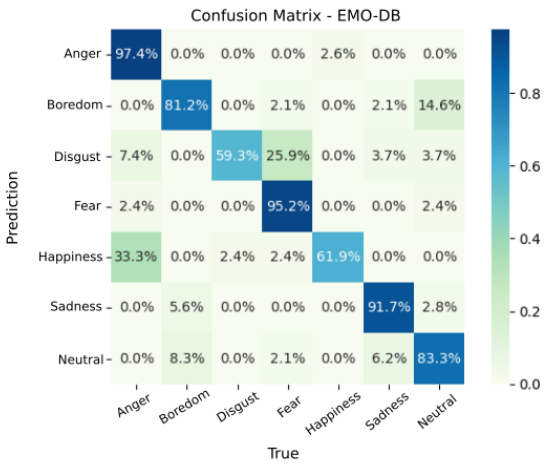
a. Test configuration

Audio files have been split using stratification and considering the amount of 80% for training and 20% for the test. In order to prevent overfitting, the mechanism of Early Stopping is employed on training data. Empirically, was defined an amount of 15 epochs as a marker. After this number, if the neural network does not improve accuracy under the validation set, the training is stopped and the best configuration is saved. Each emotion dataset has been trained 3 times. We averaged the following metrics: confusion matrix, UA, ROC-AUC, Micro, and F1-Scores.

b. Results without Sliding Window

We organized the experiments by first presenting the results of the classification algorithm without the proposed data augmentation mechanism. The objective here is to show the neural network generalization capabilities and general performance using the MFCC features as input.

Fig. 4: Confusion matrix resulting from training the EMO-DB dataset without the Sliding Window.



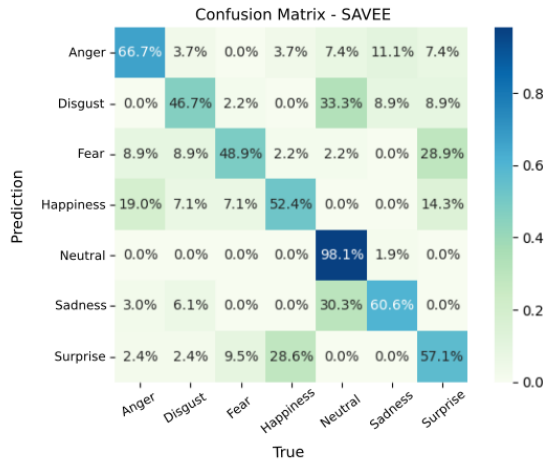
In Fig. 4 it is shown the confusion matrix of the classifier

TABLE 3: SET OF PARAMETERS USED IN THE CLASSIFICATION ALGORITHM.

Layer	Output size	Kernel Size	Stride
LFLB 1 - Convolution	128 x 256 x 64	3 x 3	1 x 1
LFLB 1 - Max Pooling	64 x 125 x 64	2x2	2x2
LFLB 2 - Convolution	64 x 125 x 64	3 x 3	1 x 1
LFLB 2 - Max Pooling	16 x 31 x 64	4 x 4	4 x 4
LFLB 3 - Convolution	16 x 31 x 128	3 x 3	1 x 1
LFLB 3 - Max Pooling	4 x 7 x 128	4 x 4	4 x 4
LFLB 4 - Convolution	4 x 7 x 128	3 x 3	1 x 1
LFLB 4 - Max Pooling	1 x 1 x 128	4 x 4	4 x 4
LSTM	256	-	-
Softmax - Dense	7	-	-

using the EMO-DB. It can be seen that some emotions such as Fear and Sadness achieved a high percentage of recognition, even with a lower amount of data analyzed than the others belonging to the EMO-DB base. Although other emotions such as Disgust and Happiness, obtained much lower results, this behavior might be explained by the lack of sufficient data for the algorithm to be able to precisely recognize the differences in such emotions. As a general result, we observe that Happiness maintains a low percentage of recognition in all datasets, as can be seen in Figs.4, 5 and 6. In addition to the number of audio files, what causes this occurrence is the fact that the algorithm is often confused with more intense emotions, such as Anger and Surprise for example.

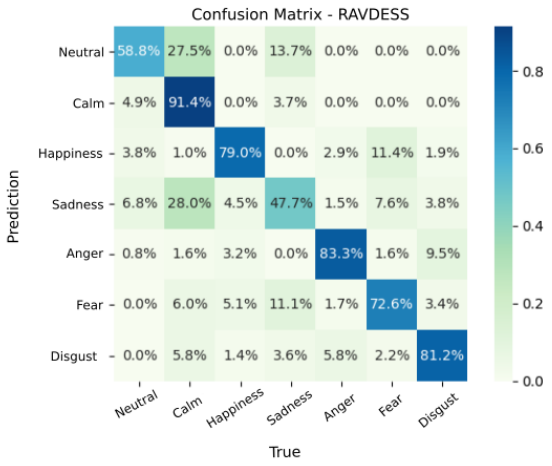
Fig. 5: Confusion matrix resulting from training the SAVEE dataset without the Sliding Window.



Observing the experiments without the Sliding Window, it is important to observe the performance variation involving the same emotion throughout different datasets. An interesting case occurs with emotion Disgust shown in Figs. 5 and 6. Even having the same number of samples as the other emotions for both datasets (as can be seen in Table 2), the discrepancy of performance values is still large.

Table 4 presents the final results of the experiments without the sliding window mechanism. It can be observed a difference of practically 20% of the recognition between the EMO-DB and SAVEE datasets, when comparing the UA function. The dataset SAVEE, which has the smallest number of audio sample, obtained the worst results in training.

Fig. 6: Confusion matrix resulting from training the RAVDESS dataset without the Sliding Window.



c. Results with Sliding Window

Figs. 7, 8 and 9, present the appropriate confusion matrices generated by the experiments in each dataset. In contrast to the confusion matrices in Subsection b, it appears that there has been a reduction in the disparity of results between emotions, such as the disgusting and neutral emotions of the SAVEE dataset in Fig. 5 there is a difference of recognition of 51.4 %. However, the corresponding results with the Sliding Window in Fig. 8 show a much smaller difference, with a percentage of 17.7 %. This evolution is due to the increase in audio samples that the algorithm produced, this made it better to calculate the weights of its layers, thus leading to a better classification capacity.

Due to the increase in data samples being different in each dataset, it implied distinct factors of growth in the final emotions recognition. In Table 5 it is shown the results obtained with the Sliding Window with the evaluated performance functions, having UA as the main observation parameter, and in Table 2 the recognition increase in each dataset is noticeable. In EMO-DB fewer audios were generated than the other datasets (as can be seen in Table 1), which led to a smaller increase in its final performance.

The promising results obtained with the Sliding Window mechanism, reinforced the idea that the increase in the amount of audio samples had led to the improvement of the algorithm's capacity to classify the input emotions.

TABLE 4: EVALUATION FUNCTIONS GENERATED FROM THE EXPERIMENTS WITHOUT THE SLIDING WINDOW.

Model	Input	Dataset	ROC-AUC	F1-Score Macro	F1-Score Micro	UA
Without SW	MFCC	EMO-DB	98.19%	82.50%	84.50%	**81.44%**
Without SW	MFCC	SAVEE	90.34%	61.1%	62.5%	**61.50%**
Without SW	MFCC	RAVDESS	95.29%	72.21%	73.60%	**73.44%**

SW = Sliding Window.

Fig. 7: Confusion matrix resulting from training the EMO-DB dataset with the Sliding Window.

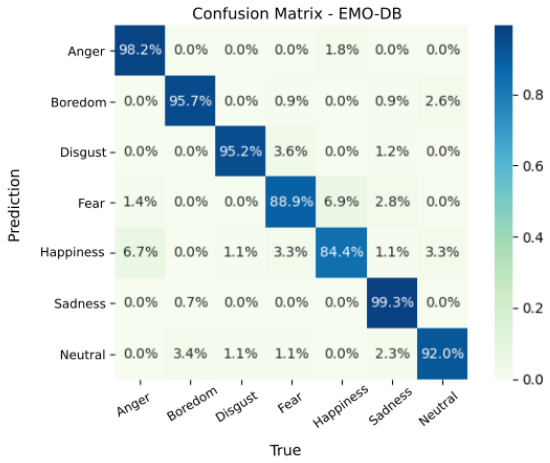


Fig. 8: Confusion matrix resulting from training the SAVEE dataset with the Sliding Window.

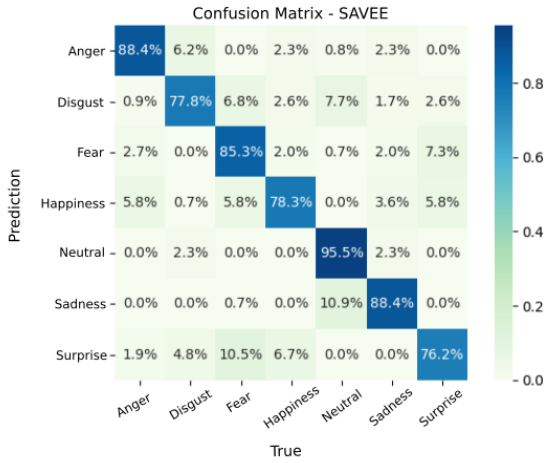


TABLE 5: EVALUATION FUNCTIONS GENERATED FROM THE EXECUTION OF THE DATASETS WITH THE SLIDING WINDOW.

Model	Input	dataset	ROC-AUC	F1-Score Macro	F1-Score Micro	UA
With SW	MFCC	EMO-DB	99.70%	93.64%	94.44%	**93.9%**
With SW	MFCC	SAVEE	98.21%	84.61%	86.00%	**84.26%**
With SW	MFCC	RAVDESS	99.52%	92.39%	92.38%	**92.26%**

SW = Sliding Window.

d. Comparison with related works

Tables 7, 8 and 9, show the comparison of the results between external works and the proposed methodology. In [1], the authors trained the classification algorithm with the audios of all the actors, except one. Therefore, the audios of this selected actor are not included in the training, and are used only in the test. This approach is known as LOSO (Leave one Speaker Out).

Fig. 9: Confusion matrix resulting from training the RAVDESS dataset with the Sliding Window.

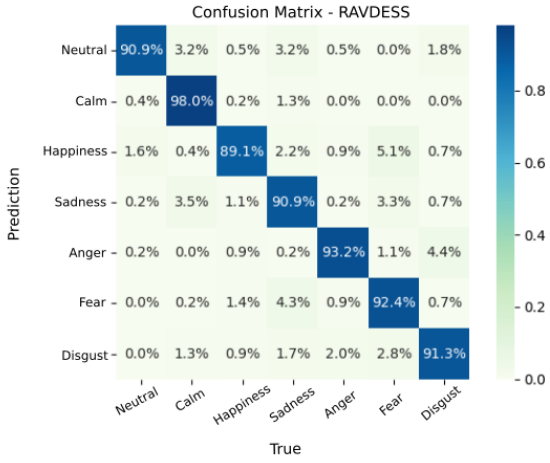


TABLE 7: COMPARISON OF EMO-DB RESULTS WITH RELATED WORKS

Work	Training method	UA
[1]	LOSO	**84.53%**
[4]	Speaker Dependent	**95.02%**
Proposed	Speaker Dependent	**93.39%**

TABLE 8: COMPARISON OF SAVEE RESULTS WITH RELATED WORKS

Work	Training method	UA
[1]	LOSO	**59.40%**
Proposed	Speaker Dependent	**84.26%**

TABLE 9: COMPARISON OF RAVDESS RESULTS WITH RELATED WORKS *THIS IS THE RESULT DISREGARDING THE SURPRISE EMOTION, PRESENT IN [3] BUT NOT IN THIS WORK.

Work	Training method	UA
[3]	Speaker Dependent	***76.85%**
Proposed	*Speaker Dependent*	**92.26%**

The results generated by this work managed to obtain an excellent performance in relation to the aforementioned works, except for [4]. As can be seen in Table 7, the work in question still managed almost 2% more in the UA assessment.

V. FINAL REMARKS

Simulating the process of recognizing human emotions on a computer, as already mentioned, is a very complex process. In addition to each emotion, there are cultural variations that can lead to different intonations. Such differences

may be very hard to capture during the classification process. This work presented a method of data augmentation of the audio samples called the Sliding Window. The algorithm used by this work, together with the Sliding Window method as pre-processing, managed to increase the accuracy of the emotions recognition in the EMO-DB dataset by 11.95%, in SAVEE by 22.76%, and RAVDESS at 18.82%. The results also have shown that the proposed method is competitive with state-of-the art approaches. The future investigations include the exploration of the window size parameter in order to determine the sensibility of the classifier over this parameter.

Having a consistent emotion recognition algorithm, which can discern the emotion of incoming audio, can be very useful in daily tasks and in more specific applications. Bearing this in mind, the application of the Sliding Window or similar method may be useful for future works hopping to achieve even more precise classification results.

REFERENCES

[1] P. Jiang, H. Fu, H. Tao, and P. L. L. Zhao, "Parallelized convolutional recurrent neural network with spectral features for speech emotion recognition," *IEEE Access*, vol. 7, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8756261

[2] Z. W. Fen Xu, "Emotion recognition research based on integration of facial expression and voice," *2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, vol. 7, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8633129

[3] S. K. Mustaqeem, "A cnn-assisted enhanced audio signal processing for speech emotion recognition," *Sensors*, vol. 20, p. 183, 12 2019.

[4] J. Zhao, X. Mao, and L. Chen, "Speech emotion recognition using deep 1d & 2d cnn lstm networks," *Biomedical Signal Processing and Control*, vol. 47, pp. 312 – 323, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1746809418302337

[5] M. A. Hossan, S. Memon, and M. A. Gregory, "A novel approach for mfcc feature extraction," in *2010 4th International Conference on Signal Processing and Communication Systems*, 2010, pp. 1–5.

[6] N. Dave, "Feature extraction methods lpc , plp and mfcc in speech recognition," 2013.

[7] S. A. Alim and N. K. A. Rashid, "Some commonly used speech feature extraction algorithms," in *From Natural to Artificial Intelligence*, R. Lopez-Ruiz, Ed. Rijeka: IntechOpen, 2018, ch. 1. [Online]. Available: https://doi.org/10.5772/intechopen.80419

[8] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp. 338–342, 01 2014.

[9] M. Matsugu, K. Mori, Y. Mitari, and Y. Kaneda, "Subject independent facial expression recognition with robust face detection using a convolutional neural network," *Neural Networks*, vol. 16, no. 5, pp. 555 – 559, 2003, advances in Neural Networks Research: IJCNN '03. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608003001151

[10] S. Khan, H. Rahmani, S. A. A. Shah, M. Bennamoun, G. Medioni, and S. Dickinson, *A Guide to Convolutional Neural Networks for Computer Vision*, 2018.

[11] P. Jackson and S. ul haq, "Surrey audio-visual expressed emotion (savee) database," 04 2011.

[12] S. R. Livingstone and 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*, vol. 13, no. 5, pp. 1–35, 05 2018. [Online]. Available: https://doi.org/10.1371/journal.pone.0196391

[13] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier, and B. Weiss, "A database of german emotional speech," vol. 5, 01 2005, pp. 1517–1520.

[14] Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, "librosa: Audio and Music Signal Analysis in Python," pp. 18 – 24, 2015.

[15] L. Rice, E. Wong, and J. Kolter, "Overfitting in adversarially robust deep learning," 02 2020.

[16] E. Lashgari, D. Liang, and U. Maoz, "Data augmentation for deep-learning-based electroencephalography," *Journal of Neuroscience Methods*, vol. 346, p. 108885, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165027020303083