Speech Emotion Recognition Using Machine Learning Techniques: A Dual Approach

Ana Filipa Gomes
FEUP
up202103356@up.pt

Pedro Pereira
FEUP
up202103372@up.pt

Ruben Silva
FEUP
up202103374@up.pt

***Abstract*—Speech Emotion Recognition is a challenging topic in data science taking part in a wide range of applications, from improving diagnostic tools to on board vehicle driving systems. In this report, we used a collection of four datasets with a total of 12162 audio samples from 8 different emotions: angry, calm, disgust, fear, happy, neutral, sad and surprised. To aboard this topic, two approaches were explored: a traditional manual feature extraction and classification as well as a deep learning approach. The first method resources to the extraction of mel-frequency cepstral coefficients, mel-scale spectrogram, chromagram, spectral contrast features, and Tonnetz representation from the audio files and uses them as inputs. Having the features extracted, we used an SVM classifier and studied the performance with and without augmentation, achieving an accuracy of (76.1 ± 1.2)% and (65.3 ± 0.6)%, respectively. The second proposed method uses mel-spectrograms images and deep convolutional neural networks (CNN) to classify the 8 emotions. Still in this approach, we compared the performance between a custom 4 layers CNN model and the VGG16 network using transfer learning, achieving an accuracy of (68,7 ± 0,8)% and (65,9 ± 1,2)%, respectively.**

***Index Terms*—Speech Emotion Recognition, Feature Extraction, Support Vector Machine, Convulotional Neural Network**

I. Introduction

The most natural way to communicate and express feelings, opinions and thoughts is through speech. In this technologically-driven world, researchers have been working on ways to replicate it in human-machine interactions since it is the most effective mean of communication among human beings. However, machines incapability to understand the persons' emotional state hidden behind their speech has been a challenge [1]. Despite the improvement in emotion recognition over the last years, for example, with the development of voice-based virtual assistants, there is still a long way to determining underlying emotions from the speakers audio signals [2]. Speech Emotion Recognition (SER) comprises the recognition of the speaker emotional state by identifying specific characteristics in their speech and can improve the interaction between humans and machines in several areas. Some applications include call centre conversations, onboard vehicle driving systems, diagnostic tools by emotion patterns analysis, and determination of situational seriousness in emergency call centres [1, 3]

A SER model must be able to identify primary emotions such as anger, surprise, happiness, calm, sadness, fear, neutral and disgust. For this reason, considerable research has been invested in the development of robust machine learning algorithms to accurately classify these emotions [4]. In this paper, two different approaches will be explored: a traditional machine learning model and a deep learning approach. A traditional SER process comprises three main stages: preprocessing, feature extraction and classification/recognition. The first stage is responsible for removing background noise, whereas the second summarizes the most relevant information in the signal resourcing to a defined number of attributes, being, therefore, correlated to the accuracy of the emotion recognition [5]. The main advantage of the deep learning model is that in this last technique the feature extraction step is automatically done. There are several deep learning methods that have succeeded in SER applications, e.g. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM), however, the most common approach consists on the combination of two of the these three mentioned before. For our specific case, we will focus on studying only CNN architecture to perform SER based on spectrogram images [1].

II. Dataset Overview

The dataset used in this paper consisted of a combination of four speech emotion datasets: the Crowd-sourced Emotional Multimodal Actors Dataset (CREMA-D) [6], Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [7], Surrey Audio-Visual Expressed Emotion (SAVEE) [8] and Toronto Emotional Speech Set (TESS) [9]. The CREMA-D was chosen due to its wide variety as it includes 7442 audios of 12 sentences from 48 male and 43 female actors with ages from 20 to 74 and a variety of races and ethnicities (African, American, Caucasian, Hispanic). The RAVDESS contains 1440 files of two statements from 12 male and 12 female professional actors with a neutral North American accent and was selected for its great availability. The SAVEE recorded 4 native English male speakers aged 27 to 31 years vocalizing 15 sentences per emotion resulting in 120 utterances per speaker. The TESS includes 2800 audio samples of 200 target words in a sentence spoken by 2 actresses aged 26 and 64. Each sentence was recorded for every emotion at study.

The described datasets were merged into one in order to have a more representative dataset that could be assessed throughout the development of the two proposed methods. This resulted in a total of 12162 audio files with the distribution presented in Table I.

TABLE I
DATASET

	CREMA-D	RAVDESS	SAVEE	TESS	Total
Angry	1271	192	60	400	1923
Calm	0	192	0	0	192
Disgust	1271	192	60	400	1923
Fear	1271	192	60	400	1923
Happy	1271	192	60	400	1923
Neutral	1087	96	120	400	1703
Sad	1271	192	60	400	1923
Surprised	0	192	60	400	652
Total	7442	1440	480	2800	12162

III. Traditional Approach

The first approach to this problem was based on traditional machine learning methods that are in greater number in the literature. In the following section we will discuss the three primary steps of a traditional classification method (preprocessing, feature extraction, and classification) and, in addition, the importance of data augmentation comparing the results of the classification with and without it.

A. Data preprocessing and augmentation

The database on which this entire project is based contains audios where there is no presence of noise so no specific preprocessing has been carried out in this respect. We need to read the audios and set a common duration to standardize the entire subsequent process of extracting features. We found that the first 0.5 seconds are mainly silent and that the majority of the information is concentrated between 0.5 and 3 seconds, so we set an offset of 0.5 seconds and a duration of 2.5 seconds.

Before feature extraction, we decided to follow two different paths to evaluate the importance of the amount of data in which the entire classification algorithm is developed. The key to getting better models is data, the more examples the model has to learn from, the better it will be able to recognize which differences in speech matter and which do not. One way to increase the data without having truly new data is to use the samples that we already have and introduce small variations on them that are not characteristic of the emotion itself, preserving the class, also known as data augmentation.

To do this, each audio has been modified in 3 different ways: adding noise, stretching the audio, and modifying the pitch - Figure 1. The first variation consisted of adding noise with uniform distribution and amplitude of 0.035. The addition of noise is not harmful as long as the noise is not very strong. The second modification consisted of a stretch, making the speech slower and longer, but likewise preserving the emotion. Finally, we have also added a modification to the pitch, changing the tone. All of these processes were done using the python *Librosa* library.

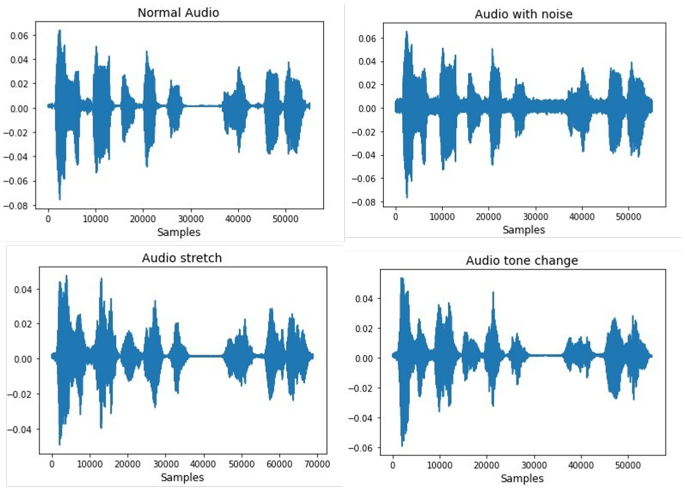

Fig. 1. Different data augmentation visualizations

B. Feature Selection

Feature extraction is a key element in machine learning models since it significantly influences the quality of the training process. Resourcing to *Librosa* audio library, the following spectral representations were used for each audio: Mel-frequency Cepstral Coefficients (MFCCs), Mel-scaled spectrogram, Chromagram, Spectral contrast feature and Tonnetz representation [10].

The MFCC and the Mel-scaled spectrogram try to simulate the human reception pattern of sound frequency and, for that reason, are the most commonly used in SER [10]. A spectrogram is the visual representation of signal strength over time at different frequencies present in a certain waveform, in a two-dimensional graph where the intensity of the colour of a point is given by the frequency amplitude. This is obtained by applying the Fast Fourier transform to the speech signal [1, 11]. MFCCs give rise to a representation of the short-term power spectrum of sound. Despite these two methods having a high success rate in identifying and tracking timbre fluctuations, this is not verified in distinguishing pitch classes and harmonies [10].

On the other hand, Chromagrams tend to do good in these last points and can be acquired resourcing to the short-time Fourier transform.

Spectral Contrast extracts the spectral peaks, valleys, and their differences in each sub-band. Thus, Spectral Contrast feature represents the relative spectral characteristics [12].

The Tonnetz measures the tonal centroids of a sound in a six-dimensional pitch space and evidence that pitch classes with close harmonic relations have smaller Euclidean distances on the plane. Analogously to the dataset mixing, the resource to different audio features enriches the description of an audio file and, therefore, the performance of SER models.

C. Feature Extraction

Taking this into account, for each audio, we extract all the features mentioned above. Starting with a slightly more detailed analysis, each feature is composed of an array with N-rows and C-columns. All features have the same number of columns as they relate to the time intervals in which each audio was broken out during the 2.5s. However, depending on the nature of each of the 5 features, these may vary in terms of the number of rows. This way, we averaged each row for each feature and saved it in a vector. With this, we can then horizontally group all the vectors thus forming the vector with the total number of attributes for each audio. The final result was 173 attributes for each audio sample.

This process was elaborated per each of the audios and each vector corresponding to audio was concatenated vertically with all the others. So our final dataset with no data augmentation is an array of [12162 x 173] in which the rows relate to the samples and the columns to their respective features. In the case where we used data augmentation, we will have 4 times more samples so the array will be [48648 x 173].

D. Training Setup

With the features extracted and our dataset defined, we can now design our model. There are several classic models and algorithms for classification in machine learning but our attention has fallen over the Support Vector Machine model (SVM) since it has proven to achieve good results in SER [3]. We tested different kernel functions - poly and RBF - and hyperparameters - C={4,7,8,9,15,25,30,50,100,200} - and found that the RBF kernel with C=100 was the one that best performs regarding our data with augmentation and, the RBF kernel with C=30 was the most suited one for the data without augmentation. However, to fully access our model performance we have also used k-fold validation. In this case, stratified k-fold was used as we want to have the same representation of each class in every fold. We choose k=5 as it is a number that allows us to test the model's performance on different (random) training and test folds while not compromising too much time on doing so. With k=5, each fold will have a random training set containing 80% of all the data and a test set with 20% of the data.

E. Traditional approach results and discussion

Using the SVM with the hyperparameters mentioned above, we validated the model for the five folds and by calculating the mean accuracy for every test fold, we achieved an accuracy of (76.1 ± 1.2)% with data augmentation and (65.3 ± 0. 6)% without any data augmentation. Thus, there was a significant increase - about 11% - when we augmented the data, emphasizing the benefits of using more data in the process of learning and generalizing.

As we are dealing with a multi label classification problem, we have also acquired the confusion matrix of each fold, registering the Precision, Recall and F1-score, which were summarized in Table II and Table III being the first referent to the results regarding data with augmentation, whereas the second to the results concerning data without augmentation.

TABLE II
STRATIFIED 5-FOLD MEAN RESULTS FOR SVM WITH AUGMENTATION

Metric **Class**	Precision	Recall	F1-score	Samples
Angry	80.5 ± 1.1	85.5 ± 0.5	83.0 ± 0.7	1538.4
Calm	83.6 ± 2.0	91.0 ± 2.1	87.2 ± 1.9	153.6
Disgust	69.5 ± 0.7	70.9 ± 0.9	70.2 ± 0.6	1538.4
Fear	74.9 ± 1.1	68.9 ± 1.4	71.7 ± 1.2	1538.4
Happy	76.8 ± 1.8	71.4 ± 0.8	74.0 ± 0.8	1538.4
Neutral	75.8 ± 0.9	75.2 ± 0.9	75.5 ± 0.5	1362.4
Sad	72.2 ± 0.6	77.6 ± 1.6	74.8 ± 1.1	1538.4
Surprised	94.2 ± 0.7	91.3 ± 1.2	92.7 ± 0.4	521.6

TABLE III
STRATIFIED 5-FOLD MEAN RESULTS FOR SVM WITHOUT AUGMENTATION

Metric **Class**	Precision	Recall	F1-score	Samples
Angry	73.1 ± 2.0	76.1 ± 1.8	74.6 ± 1.5	384.6
Calm	61.9 ± 1.7	75.5 ± 8.4	67.9 ± 4.1	38.4
Disgust	58.9 ± 2.0	58.2 ± 1.2	58.5 ± 1.4	384.6
Fear	63.6 ± 2.8	54.8 ± 3.3	58.8 ± 2.4	384.6
Happy	62.9 ± 1.1	57.6 ± 0.6	60.1 ± 0.4	384.6
Neutral	62.6 ± 1.3	66.1 ± 1.6	64.3 ± 1.4	340.6
Sad	63.4 ± 1.0	71.6 ± 1.8	67.2 ± 1.3	384.6
Surprised	87.2 ± 3.4	84.7 ± 1.6	85.9 ± 2.3	130.4

For this specific problem, there are no costs related to wrongly predicting a specific emotion, so overall we want to understand if our model is able to correctly differentiate one class from another. An high class precision tells us that the model is good at minimizing the false positive rate, which means that he rarely misclassifies if he says that it is positive. On the other hand, high recall tells us that the model is good at minimizing the false negative rate, which means that out of actual positive data, he rarely misclassifies the emotion. As one case is not preferable to the other for this multi-label problem, we will also took into account the F1-score, as it considers both precision and recall.

By analysing the tables we can verify that for both cases the easiest emotion to be recognized is ”surprised” with a F1-score of (92.7 ± 0.4)% and (85.9 ± 2.3)% with and without data augmentation, respectively. On the other hand, the most difficult emotion to be predicted is ”fear” with a F1-score of (71.7 ± 1.2)% and (58.8 ± 2.4)%, respectively. These results are easily explained from a human point of view, due to the nature and characteristics of the emotion itself, for example, the speech of someone surprised can have more specific attributes than the speech of someone with fear which can be more prompt to confusion with other emotions. Alongside the ”fear”, the ”disgust” emotion is also presented as though one to differentiate. Concerning the ”calm” emotion, even though having a considerable less number of samples, both models were able to provide decent results. However, for the emotion ”calm”, the model that was trained with augmentation showed a significantly higher improvement when compared with the other classes, which once again emphasise the augmentation effect. It is also very important to point out the standard deviation values of the metrics from one case to the other, as they were, for most of the classes, reduced in the augmentation model.

Concluding and given that we are working with multi class classification, we can consider that our results are satisfactory and according to those present in the literature, being the augmentation model overall superior when compared with the one without augmentation.

IV. Deep learning approach

In the following section we will cover in depth the deep learning models that were used, how they were designed, what results did we manage to achieve and draw a comparison between them.

A. *Data preprocessing*

For the deep learning approach, we have decided to work with CNN as different studies [1, 3] have proven they were quite able of learning crucial features from spectrogram-based images. Thus, in this case, our input will be the Mel-spectrogram images extracted from the audio samples. The Mel-spectrogram *Librosa* function takes the same parameters as before but instead of doing the mean over all rows, we save the plot figure as a RGB image - Figure 2. No further image data augmentations were performed due to memory constraints.

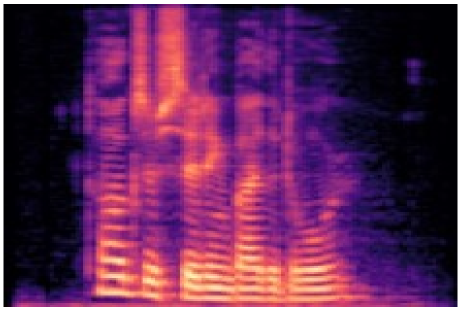

Fig. 2. Example of a Mel-Spectogram

Then we normalized the images by dividing them by 255 and we loaded them from the file with the help of the *keras.preprocessing* method which allows us to set a specific size - which we choose to be 128x128 - and a specific color mode which we set as RGB. Another important step is one-hot encoding the labels as the vast majority of deep learning algorithms are built to operate with numeric values as an efficiency constraint.

B. *VGG16 model - transfer learning*

As it's common in deep learning problems, we have started by using a very well know deep learning network, the VGG16, which is a CNN that was proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition” [13]. Figure 3 displays the model structure in detail.

It was chosen to not include the fully-connected layer at the output end of the architecture (identified in Figure 3 with green) by setting “include_top = False”, but rather to separately add the remaining layers - Flatten and Dense Output Layer - to the VGG16 model. We have also included a Dropout Layer later on with the value of 0.5 to curtail the overfitting of the model. Therefore, it was possible to specify the input shape as wanted (128, 128, 3) and, as we don't want to train all 138 million parameters that the model contains but instead apply transfer learning, we set the weight parameter as ”imageNet” which allows us to use the pre-trained weight gathered from training on the ImageNet dataset.

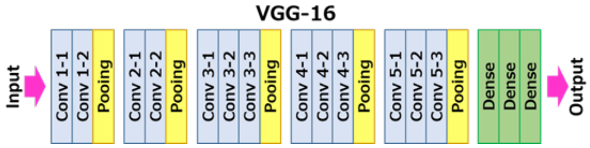

Fig. 3. VGG16 model architecture

C. *Custom CNN model*

For comparison, a custom CNN with four convolutional layers was built based on different Kaggle notebooks and articles [1].

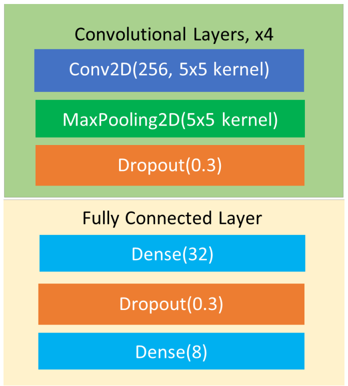

Fig. 4. CNN custom made structure

The first layer prepare the model to receive data (input) with a certain size that is beforehand specified. Then we have 4 blocks of convolutional layers followed by a max pooling, as represented in the Figure 4. As we are dealing with image data with a certain height and width, our convolutional and max pooling layers will all be 2D. The convolutional layers are responsible for extracting the features from the spectrogram, and this is done by applying filters to the image.

The filter number per block is 256 in the first two blocks, decreasing then to 128 in the third and 64 in the fourth block. The filter size is an important parameter, and in this case, 5x5 kernel size was applied for each convolutional layer, with strides of 1 and padding set as ”same”. For the activation functions we used ReLU as a way to address the phenomenon known as “vanishing gradients”.

The max-pooling layers reduce the dimensions of the input image by combining the outputs of a grid of pixels at one layer into a single neuron in the next layer. In this case, it chooses the highest intensity pixel in one cluster of 5x5 and passes it on to the next layer, therefore being called max-pooling. Two dropout layers of 0.3 were also included in the model as they help reduce the overfitting by turning off random neurons and adding some noise in the process. This way, the situation where some neurons are highly responsible for the output prediction is lessened.

After all the convolution blocks, we need to flatten the features extracted and pass them to the last dense layer that contains the activation function that will be used to obtain the model outputs. In this case, we used a Softmax with 8 units (one for each class) for classification. This model was trained from zero with a total number of 2,813,672 trainable parameters.

D. Training setup

As we are dealing with a multi-label classification problem, we used the categorical cross-entropy loss function and as our optimizer the adaptive moment estimation (Adam) which is computationally efficient and requires little memory. As a better way to access our model performance, we used k-fold validation, with k=5 as previously done in the traditional approach. The two deep learning models were trained using Google Colab and Kaggle GPU support, for a total of 100 epochs, with a batch size of 32 and a learning rate of 0.00005. These hyperparameters were selected after testing different ones, such as batch sizes of 16, 64, 128, and learning rates of 0.05 0.001, and 0.0005. As its common to observe overfitting, we have also defined an early stopping callback method in which if the loss value of the validation set doesn't improve (meaning it doesn't reduce) after 10 epochs the training stops, and the best weights are reset. This will prove very useful, as we will discuss later.

E. Deep Learning approach results and discussion

Concerning the Keras VGG16 transfer learning model, with imagenet initial weights, after compiling and applying the mean over the five folds, the test results show an overall accuracy of (65,9 ± 1,2)%. The learning curve presented in Figure 5 is a random one selected from the five curves gathered over the folds as all of them have a very close behavior. We can notice that the validation curve starts higher and only around the 7 epoch is passed by the training curve. We consider that this unexpected phenomenon is due to the dropout layers disabling some neurons, which make training less prompt to overfitting but also, in a way, more difficult. However, in validation, the model has access to all neurons so it may perform better in the beginning. Around the 15 epoch the training and validation curves start to diverge, and the model starts overfitting. Then, the model proceeds to stop at 30 epochs as the minimum value of the loss doesn't improve after the 20th epoch.

When it comes to the CNN custom model, after compiling and applying the five folds we manage to achieve an accuracy of (68,7 ± 0,8)%. The learning curve presented in Figure 6 is a random one selected from the five curves gathered over the folds. We can notice the same effect regarding the transfer learning model when it comes to the validation accuracy starting with higher values. Still, the training and validation curves behave very closely which means that the model is both learning and generalizing until around the 35 epoch , where the model is no longer able of generalize and starts overfitting which triggers the early callback around the 55 epoch.

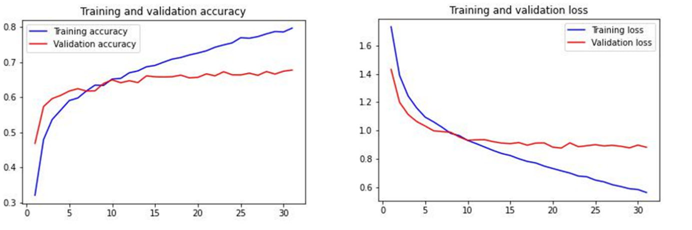

Fig. 5. Learning curves of the transfer learning model for both loss and accuracy

Overall, the custom model takes longer to train, which makes sense as it needs to learn a higher number of parameters and besides that the VGG16 transfer learning model starts overfitting earlier. When it comes to overall accuracy, the custom model presents better results as expected from the learning curves.

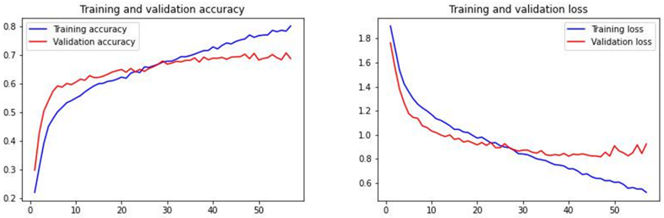

Fig. 6. Learning curves of the custom model for both loss and accuracy

However, analyzing only the overall accuracy and the learning curves does not provide enough information about the model performance for a particular class, as discussed in section III-E.

As in the traditional method, we have also acquired the confusion matrix of each fold, registering the Precision, Recall and F1-score, which were summarized in Table IV and Table V being the first referent to the results regarding the transfer learning model, whereas the second to the results concerning custom model.

By analysing the tables we can verify that for both cases the easiest emotion to be recognized is ”surprised” with a F1-score of (85.4±1.3)% and (86.8±1.8)% for transfer learning model and custom model, respectively. On the other hand, the most difficult emotion to be predicted is ”fear” with a F1-score of (59.3±2.0)% and (63.6±0.7)%, respectively. Also, the ”calm” class portraits a relatively low Recall value in the custom model (56.8±6.0)%, being this model more affected by the unbalance data when compared with the transfer learning one (79.2±12.2)%. It is also worth mentioning that the class ”angry” presents very close values for both Precision and Recall, in the two models, which shows that they are quite capable of differentiating this class from the others. Concerning the standard deviation of the metrics, both models present similar values so there was no significant difference to report.

TABLE IV
Stratified 5-Fold mean results for VGG16 transfer learning model results

Metric Class	Precision	Recall	F1-score	Samples
Angry	73.1±2.3	75.5±2.5	74.2±1.4	384.6
Calm	61.1±7.8	79.2±12.2	67.9±4.0	38.4
Disgust	64.2±4.6	60.7±3.3	62.2±2.3	384.6
Fear	62.1±2.8	56.9±2.6	59.3±2.0	384.6
Happy	64.1±4.9	58.5±7.0	60.6±2.1	384.6
Neutral	66.2±4.2	71.0±5.8	68.2±2.3	340.6
Sad	62.2±4.6	64.5±5.1	63.0±1.3	384.6
Surprised	83.4±3.6	87.7±4.6	85.4±1.3	130.4

TABLE V
Stratified 5-Fold mean results for custom CNN model

Metric Class	Precision	Recall	F1-score	Samples
Angry	77.9±2.9	77.4±2.2	77.6±2.0	384.6
Calm	64.4±10.8	56.8±6.0	59.8±6.1	38.4
Disgust	68.4±3.6	59.1±4.5	63.2±3.0	384.6
Fear	67.5±3.0	60.4±1.6	63.6±0.7	384.6
Happy	68.9±3.6	64.7±3.1	66.6±1.0	384.6
Neutral	66.6±1.9	76.6±2.6	71.2±1.8	340.6
Sad	59.9±1.5	70.1±2.0	64.6±0.7	384.6
Surprised	86.8±2.4	87.0±3.1	86.8±1.8	130.4

No direct comparisons with literature were possible, as we could not find literature that used the same collection of datasets. In spite of that, some overall ideas can still be withdraw. For example, in [1], the authors designed a CNN model with three convolution and three fully-connected layers for SER, scoring 61.75%. They have also pointed the ”fear” class as the most difficult to differentiate.

When comparing with approaches that take into account time features, such as the one developed by [14] - which used a LSTM-CNN model - the results improve significantly, in this case being able to achieve 83.11% of accuracy. It is interesting to understand the importance that time features bring to the classifiers.

V. Conclusion

In this project, we attempt to solve the problem of SER using the traditional pipeline with SVM for classification and a second approach based on deep CNN. Inside of each of these approaches, different studies were made. The dataset used for this work consisted on the collection of four SER datasets - CREMA-D, RAVDESS, SAVEE and TESS - totaling 12162 audios from 8 different emotions: angry, calm, disgust, fear, happy, neutral, sad and surprised.

Concerning the traditional approach, we extracted five key features: MFCCs, Mel-scaled spectrogram, Chromagram, Spectral contrast and the Tonnetz representation and perform data augmentation by adding noise, stretching the audio and altering the pitch. In this first section we compared the SVM performance using the data with and without augmentation, obtaining an accuracy of (76.1 ± 1.2)% and (65.3 ± 0.6)%, respectively.

Concerning the deep learning approach, two different experiments were performed. At first, we used a pre-trained VGG16 model to determine the suitability of transfer learning in solving the problem of SER. In the second experiment we trained a custom CNN model based on the Mel-spectrograms generated from 12162 samples. The transfer learning model scored an overall accuracy of (65,9 ± 1,2)% and the custom made CNN a score of (68,7 ± 0,8)%.

In conclusion, the two approaches managed to do considerably well, having obtained over 65% accuracy for all models. Besides, the ”fear” class was the hardest one to discriminate and the ”surprised” one the easiest for the two approaches. The traditional approach with augmentation obtained the better overall performance, as oppose to expected. This can be explained by the usage of data augmentation as well as a higher variety of feature sources. It is also important to point out the difference between the standard deviation values from the k-fold validation between the two approaches, as the traditional one obtained considerably lower values, proving that the model is more consistent when testing on unseen data.

Further work is needed to improve both of the approaches. Concerning the traditional approach, we can work different feature combination as well as test different hyperparameters. On the other hand, regarding the deep learning methods, a possible solution to improve performance is the inclusion of deep learning structures that are able to learn time series features, such as LSTMs, experiment different set of hyperparameters and, another improvement could be the use of more data (using augmentation or including more datasets).

References

[1] Abdul Malik Badshah, Jamil Ahmad, Nasir Rahim, and Sung Wook Baik. Speech emotion recognition from spectrograms with deep convolutional neural network. *2017 International Conference on Platform Technology and Service, PlatCon 2017 - Proceedings*, 3 2017.

[2] Beenaa Salian, Omkar Narvade, Rujuta Tambewagh, and Smita Bharne. Speech emotion recognition using time distributed cnn and lstm. In *ITM Web of Conferences*, volume 40, page 03006. EDP Sciences, 2021.

[3] Ruhul Amin Khalil, Edward Jones, Mohammad Inayatullah Babar, Tariqullah Jan, Mohammad Haseeb Zafar, and Thamer Alhussain. Speech emotion recognition using deep learning techniques: A review. *IEEE Access*, 7:117327–117345, 2019.

[4] Yue Xie, Ruiyu Liang, Zhenlin Liang, Chengwei Huang, Cairong Zou, and Björn Schuller. Speech emotion classification using attention-based lstm. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(11):1675–1685, 2019.

[5] Sabur Ajibola Alim and N Khair Alang Rashid. *Some commonly used speech feature extraction algorithms*. IntechOpen London, UK:, 2018.

[6] Houwei Cao, David G Cooper, Michael K Keutmann, Ruben C Gur, Ani Nenkova, and Ragini Verma. Crema-d: Crowd-sourced emotional multimodal actors dataset. *IEEE transactions on affective computing*, 5(4):377–390, 2014.

[7] The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PLOS ONE*, 13:e0196391, 5 2018.

[8] S. Haq and P.J.B. Jackson. Speaker-dependent audio-visual emotion recognition. In *Proc. Int. Conf. on Auditory-Visual Speech Processing (AVSP'08), Norwich, UK*, Sept. 2009.

[9] Kate Dupuis and M Kathleen Pichora-Fuller. Browse items in this collection by the following.

[10] Dias Issa, M Fatih Demirci, and Adnan Yazici. Speech emotion recognition with deep convolutional neural networks. *Biomedical Signal Processing and Control*, 59:101894, 2020.

[11] Deepak Bharti and Poonam Kukana. A hybrid machine learning model for emotion recognition from speech signals. In *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, pages 491–496, 2020.

[12] Dan-Ning Jiang, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, and Lian-Hong Cai. Music type classification by spectral contrast feature. In *Proceedings. IEEE International Conference on Multimedia and Expo*, volume 1, pages 113–116 vol.1, 2002.

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 9 2014.

[14] Dengke Tang, Junlin Zeng, and Ming Li. An end-to-end deep learning framework for speech emotion recognition of atypical individuals. In *INTERSPEECH*, 2018.