Emotion Recognition from Speech: Comparison Between Machine Learning Classifiers

Matteo Brancatelli
Università degli Studi di Milano
Computer Science Department
Via Celoria 18, Milan, Italy
Email: matteo.brancatelli@studenti.unimi.it

***Abstract*—Recently, increasing attention has been directed to the study of emotion recognition from speech. This paper propose a system that exploit a large number of samples for training and a combined time- and frequency-domain features extraction for automatic SER. A comparision between three widely used machine learning classifiers is performed to establish which one can reach the best accuracy.**

I. Introduction

In recent years, the field of artificial intelligence has witnessed a growing interest in the recognition and interpretation of human emotions, especially through speech signals. Emotion recognition from speech, a crucial aspect of human communication, holds significant promise for a wide range of applications spanning human-computer interaction, healthcare, education, and beyond. Leveraging advancements in machine learning, signal processing, and affective computing, researchers have endeavored to develop robust and accurate systems capable of automatically detecting and interpreting emotional cues embedded within speech signals. In this paper, a speech emotion recognition system designed to address the complexities and challenges inherent in accurately identifying and categorizing emotions from spoken language is proposed. A comparison between three widely used machine learning classifiers is performed to decree which is the one that reach the highest accuracy with the extracted features.

II. Related Work

Human speech is the most common way of communication. In speech processing one of the most complex task is speech emotion recognition. It is not a trivial task since it implies that the system must understand the user's emotions. It involves analyzing various acoustic features, such as pitch, intensity, tempo, and spectral characteristics, to infer the emotional state of the speaker. SER has gained significant interest due to its potential applications in fields such as human-computer interaction (HCI), healthcare [5], customer service [11], market research, and smart home assistants [4].
There are several challenges while dealing with emotions and speech signals. Emotions are a complex and subjective phenomena, making their detection from speech signals challenging. Emotions can be influenced by cultural, contextual, and individual factors, adding further complexity to the recognition process. Also, collecting large datasets with accurately annotated emotional labels is crucial for training robust SER models. However, such datasets are often limited in size and diversity, with audio samples obtained from acted speech that not face the difficulties that a natural database faces [10].
Emotions are often influenced by the context in which they occur. Understanding contextual cues, such as the topic of the conversation or the speaker's situation, is essential for accurate emotion recognition. An improvement of the accuracy of SER systems can be obtained by combining information from multiple modalities, such as speech, facial expressions, and gestures [1][2][9].

III. Databases

In the presented work, a combination of two datasets widely used in studies related to emotion recognition are employed. Table I reports how many samples for each emotion have been used to perform the emotion recognition task.

A. RAVDESS Dataset

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) dataset is a collection of audio and video recordings designed for research in emotion and speech signal processing [7]. It contains recordings of 24 (12 female, 12 male) professional actors performing various emotions, including happy, sad, angry, fearful, surprise, and neutral expressions. The dataset is widely used in studies related to speech emotion recognition, affective computing and related fields. In this work, only happy, fearful, disgust, sad, neutral,and angry audio recordings have been considered.

B. CREMA Dataset

The Crowdsourced Emotional Multimodal Actors Dataset [3] is another widely used dataset in the field of emotional reserach, particularly in the context of speech and facial expressions recognition. It consists of audio and video recordings of actors who were instructed to perform emotional scenes, expressing a range of emotions such as happiness, sadness, anger, fear, neutral and disgust. It contains 7,442 original clips from 91 actors that spoke from a selection of 12 sentences presented using one of six different emotions and four different emotion levels (Low, Medium, High, and Unspecified).

Fig. 1. Architecture of the proposed speech emotion recognition model

	RAVDESS[7]	CREMA-D[3]	Combined
Angry	192	1271	1463
Disgust	192	1271	1463
Fear	192	1271	1463
Happy	192	1271	1463
Neutral	96	1087	1183
Sad	192	1271	1463
Total	**1056**	**7442**	**8498**

TABLE I
Number of audio samples for each emotion in the employed datasets.

IV. System Description

The proposed system consists of five main blocks. It starts with the data preparation phase, were the combined dataset is prepared and subjected to data augmentation techniques to increase the number of samples and to adapt the classifiers to real-world scenarios. The second step is feature extraction, where time- and frequency-domain features useful for the recognition are extracted from audio samples. The clustering step is not essential for classification but it is useful to visualize and analyze the feature space. In the fourth step, each of the proposed classifier is trained and, in the last step, evaluated with the help of several evaluation metrics. Figure 1 illustrates the block diagram of the proposed model.

A. Data Preparation

After combining the two speech emotion datasets, each sample has been processed using the librosa[8] python library. Each audio sample has been loaded using the `librosa.load` function with the duration parameter set to 2.5 seconds. In such manner, all audio samples have the same length. After loading all samples, a data augmentation technique has been applied to the dataset. In particular, each sample was altered with some noise, pitch shifted, and modified with both noise and pitching applied.

B. Feature Extraction

1) Time Domain Features: Time-domain audio features are usually extracted directly from the samples of the audio signal. Such features offer a simple way to analyze audio signals, although it is usually necessary to combine them with more sophisticated frequency-domain features. What follows are the time-domain features used in this work.

*a) **Zero-Cross Rate**:* is the rate of sign changes of the signal during the frame. It can be interpreted as a measure of the noisiness of a signal. It usually exhibits higher values for noisy signals and it also known to reflect the spectral characteristics of a signal. It is easy to compute and it is obtained according to equation:

$$Z(i) = \frac{1}{2W_L} \sum_{n=1}^{W_L} |sgn[x_i(n)] - sgn[x_i(n-1)]| \quad (1)$$

*b) **Root Mean Squared Energy**:* for a discrete-time signal $x_i(n)$ with W_L samples, the RMS energy (RMSE) is calculated as follows:

$$RMSE = \sqrt{\frac{1}{W_L} \sum_{n=1}^{W_L} |x_i(n)|^2} \quad (2)$$

where $x_i(n)$ is the sequence of audio samples of the ith frame with length W_L. It computes the squared root of the so-called power of the signal. This feature provides a measure of the "effective" amplitude of the signal, giving more weight to larger amplitudes. In audio processing, energy is often used to characterize the overall loudness of intensity of a signal.

2) Frequency Domain Features: Features that are based on the Discrete Fourier Transform are called frequency (or spectral) audio features. Unlike the time domain, where the signal is represented as a function of time, in the frequency domain, the signal is decomposed into its constituent frequencies, revealing the amplitude and phase information associated with each frequency. Each frequency component represents a sinusoidal wave of a particular frequency and by combining these components it is possible to reconstruct the original signal in time domain. In this work, the frequency feature. MFFCs is the frequency feature used in this work.

*a) **MFFCs**:* Mel-Frequency Cepstrum Coefficients (MFFCs) have been very popular in the field of speech processing. They are actually a type of cepstral representation of the signal, where the frequency bands are distributed according to the mel-scale. MFCCs have been widely used in speech recognition, musical genre classification, speaker clustering and many other audio applications.

C. Features Clustering

After extracting the features from the audio samples, a clustering technique known as K-Means is used in order to visualize the feature space. K-Means clustering is a widely

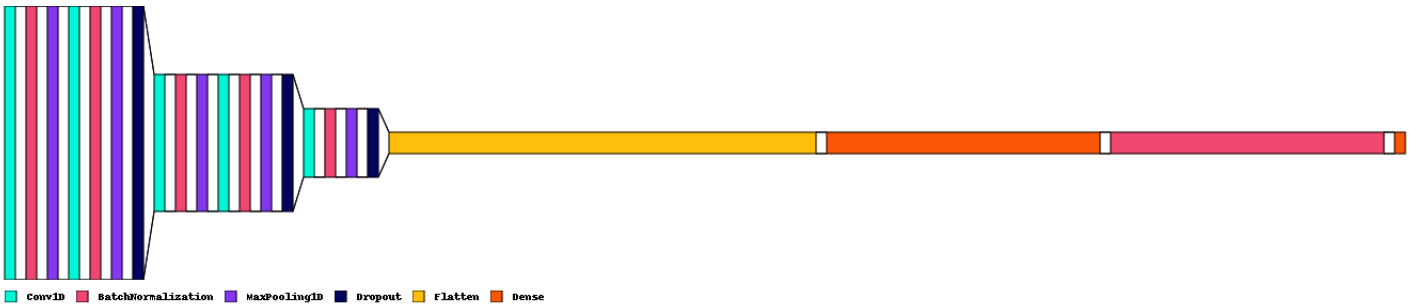

Fig. 2. Visual representation of the proposed CNN model using [6]

employed algorithm in unsupervised machine learning, designed for partitioning a dataset into distinct groups based on inherent patterns and similarities. K-Means algorithm follows an iterative approach, as shown in 1. It basically tries to reduce the within-cluster variance by iteratively assigning data points to the nearest centroid, updating them until convergence is reached. Figure 3 shows the result of KMeans clustering applied to the two principal components of the feature space.

Algorithm 1 K-Means Clustering

1: **Input:** Dataset X, Number of clusters k
2: **Output:** Cluster assignments, Centroids
3: **Initialization:**
4: Randomly select k data points as initial centroids
5: **repeat**
6: Form K clusters by assigning each point to its closest centroid.
7: Recompute the centroid of each cluster.
8: **until** Centroids do not change

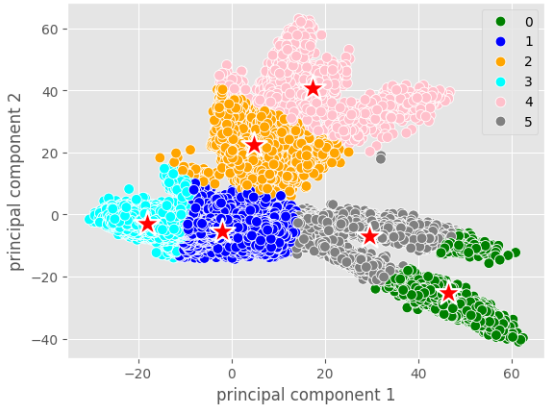

Fig. 3. KMeans clustering of the two principal components. Red stars represent the cluster centroids

D. Classification

1) ***K-Nearest Neighbors (KNN):*** The K-Nearest Neighbors classifier is a very simple classifier well suited for both binary and multi-class problems. Its main characteristic is that it does not require a training stage but rather, the training samples are used directly by the classifier during the classification stage. The main idea behind KNN is that, given a test pattern x, we first detect detect its $k-$nearest neighbors in the training set and count how many of those belong to each class. The class with the highest number of neighbors takes the test pattern x.

2) ***Support Vector Machine (SVM):*** SVM represent a powerful class of supervised learning algorithms with applications in classification and regression tasks. At the core of SVMs lies the concept of finding an optimal hyperplane that maximally separates data points belonging to different classes. This hyperplane is determined by support vectors, which are data points located near the decision boundary. Their ability to handle high-dimensional data and flexibility in dealing with non-linear relationships contribute to their widespread adoption. In this work, a One-vs-One strategy is adopted to deal with multi-class classification.

3) ***Convolutional Neural Network (CNN):*** CNNs represent a class of neural networks designed to process and analyze grid-like data, making them particularly well-suited for image and video recognition. A CNN possesses particular layers called convolutional layers. These layers perform a convolution operation using some filters on the layer's input data. The CNN exploits these convolutional layers to learn the characteristics of the input training patterns.
Figure 2 shows the proposed model structure. The network architecture is a combination of convolutional layers, maxpooling, batch normalization and dropout layers. 1D convolutional layers are used since the input of the network is the 1D features vector associated to an audio sample. Input and hidden layers use ReLu activation function while softmax function is used in the output layer. Batch Normalization layers are used to normalize the outputs of the convolutional layers.

E. Evaluation Metrics

To evaluate the performances of classification methods, several measures can be used. What follows is a brief description of the employed evaluation metrics in this work.

1) ***Confusion Matrix:*** The confusion matrix is a $N_c \times N_c$ matrix (with c number of classes), whose rows and columns refer to the true (ground truth) and predicted class, respectively. In other words, each element of the confusion matrix $CM(i, j)$, stands for the number of samples of class i that

were assigned to class j by the classification method. It follows that the diagonal elements capture the correct classification decisions.

2) ***Accuracy****:* The *overall accuracy* is defined as the fraction of samples of the dataset that have been correctly classified. It can be seen that the overall accuracy can be computed by dividing the sum of the diagonal elements of the confusion matrix by the total sum of the elements of the matrix:

$$Acc = \frac{\sum_{i=1}^{N_c} CM(i,i)}{\sum_{i=1}^{N_c}\sum_{j=1}^{N_c} CM(i,j)} \tag{3}$$

The quantity $1 - Acc$ is the overall classification error.

3) ***Recall****:* The *class recall* is a measure that describe how well the classifier performs on each class. It is defined as the proportion of data with true class label i that were correctly assigned to class i:

$$Re(i) = \frac{CM(i,i)}{\sum_{j=1}^{N_c} CM(i,j)} \tag{4}$$

4) ***Precision****:* Precision is also a class-specific measure as recall and can be defined as the fraction of samples that were correctly classified to class i if we take into account the total number of samples that were classified to that class. It is defined according to the equation:

$$Pr(i) = \frac{CM(i,i)}{\sum_{j=1}^{N_c} CM(j,i)} \tag{5}$$

5) F_1***-measure****:* This measure is computed as the harmonic mean of the precision and recall values:

$$F_1(i) = \frac{2Re(i)Pr(i)}{Pr(i) + Re(i)} \tag{6}$$

V. Experiments and Results

In this section, we describe how the classifiers used were trained and their performance. The KNN algorithm was performed using a grid-search approach: it was run with the combination of neighbors $k = 5, 10, 15, 20$ and weights functions *uniform* and *distance*. The algorithm ran in total 8 times. The best results were obtained with $k = 5$ and distance function weight with an overall accuracy of 66%. Table III shows the evaluation metrics obtained for each emotion.

	Precision	Recall	F_1-score
Angry	0.86	0.73	0.79
Disgust	0.72	0.58	0.64
Fear	0.60	0.62	0.61
Happy	0.73	0.62	0.67
Neutral	0.62	0.58	0.60
Sad	0.53	0.80	0.64

TABLE II
Evaluation metrics for KNN algorithm with $k = 5$ and distance weigth.

As it can be seen, anger emotion obtained the best f1-score with 79% and all the emotions obtained a value of at least 60%. Figure 4 reports the confusion matrix for the KNN classifier. We can observe that lots of disgust, fear, and neutral samples were misclassified as sadness. In general, the k-nearest neighbors algorithm did not perform poorly considering its simplicity.
The second classifier evaluated is the Support Vector Machine. SVM was evaluated using a grid search approach with the following hyper-parameters:

- Regularization parameter $c = 5, 10, 50, 500$
- Polynomial, linear and radial-basis kernel function
- One-vs-one decision function

The results outperformed the KNN algorithm since it obtained an overall accuracy of 89% with $c = 500$ and radial-basis function as kernel function. All the emotions have been correctly classified with a precision above the 88%. Table II reports the evaluation metrics obtained for each class. Figure 5 shows the confusion matrix for the SVM classifier. It can be seen from the confusion matrix that the misclassification of the disgust, fear, and neutral emotions are considerably reduced.

	Precision	Recall	F_1-score
Angry	0.90	0.94	0.92
Disgust	0.88	0.90	0.89
Fear	0.88	0.86	0.87
Happy	0.88	0.89	0.89
Neutral	0.92	0.86	0.89
Sad	0.90	0.89	0.90

TABLE III
Evaluation metrics for SVM algorithm with $c = 500$ and rbf as kernel function.

The last classifier evaluated is the Convolutional Neural Network. The network architecture described in Section IV-D3 has been trained for 60 epochs with a batch size of 32 and evaluated with the Train-Valid-Test split technique. The dataset has been divided in three subsets:

- **train dataset** (60% of samples): used by the model for learning, that is, to fit the parameters to the model
- **valid dataset** (20% of samples): used for provide an unbiased evaluation of a model fitted on the training dataset while tuning model hyper-parameters.
- **test dataset** (20% of samples): set of data used to provide an unbiased evaluation of a final model fitted on the training dataset

The proposed CNN model provides a very good performance with an overall accuracy of 91% on the test dataset and a training accuracy of 99.8%. In comparison with

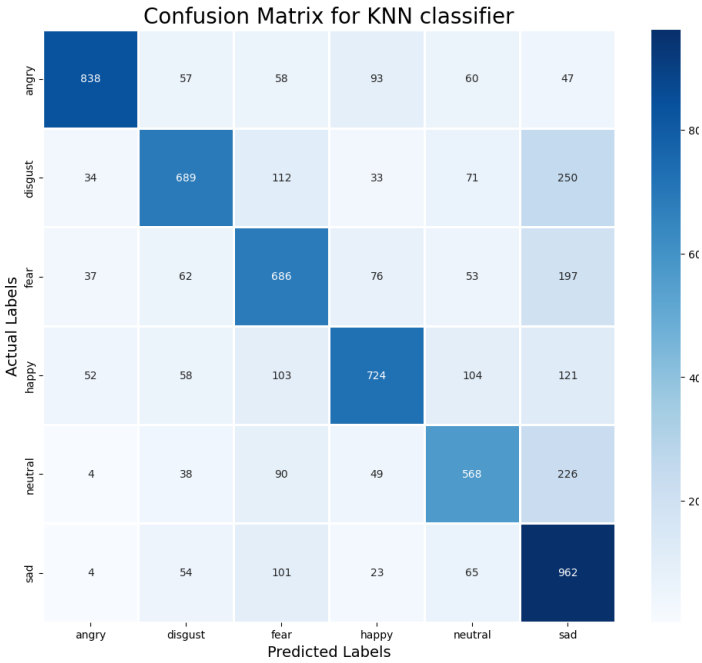

Fig. 4. Confusion matrix for the KNN classifier with $k = 5$

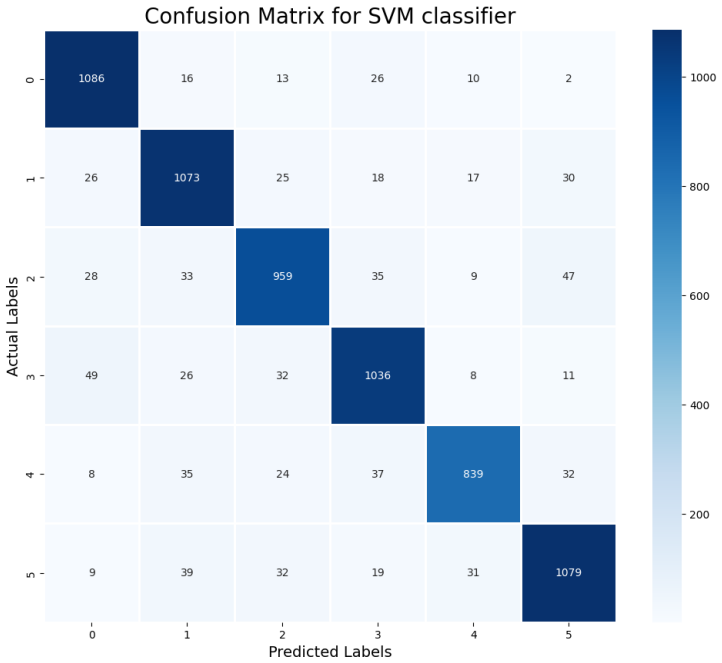

Fig. 5. Confusion matrix for the SVM classifier with $c = 500$ and rbf as kernel function.

the SVM, the results are slightly better but the training time was much higher (more than four times slower). The evaluation metrics computed for the proposed CNN are reported in Table IV and the respective confusion matrix is shown in figure 6. During the training phase, the `keras.callbacks.ReduceLROnPlateau` API was used for progressevly reduce the learning rate parameter when the accuracy stopped improving after a "patience" number of epochs equals to 3. The training stopped at 60 epochs of training because the `keras.callbacks.ModelCheckpoint` API monitored that the validation loss metric stopped improving in 5 consecutive epochs.

	Precision	Recall	F_1-score
Angry	0.91	0.95	0.93
Disgust	0.91	0.88	0.90
Fear	0.92	0.90	0.91
Happy	0.92	0.90	0.91
Neutral	0.92	0.91	0.92
Sad	0.89	0.94	0.91

TABLE IV
Evaluation metrics for the proposed CNN model.

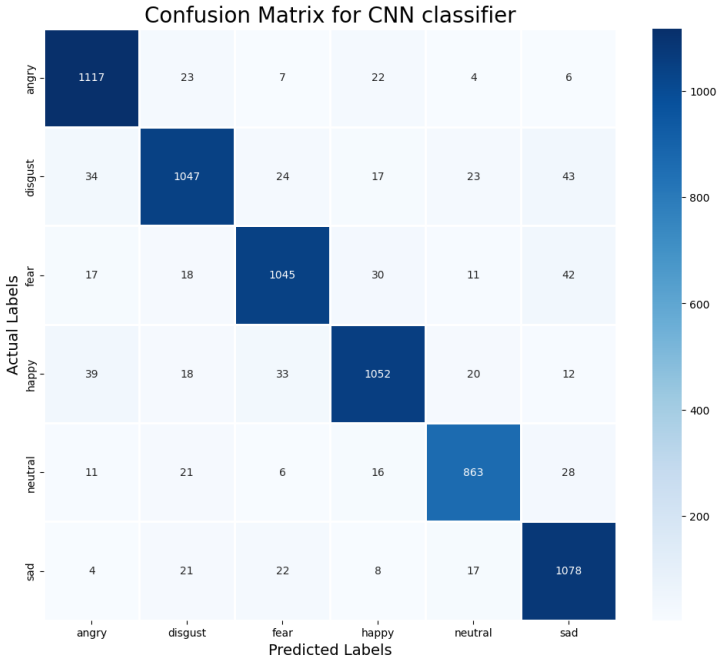

Fig. 6. Confusion matrix for the proposed CNN model.

VI. Conclusion and Possible Improvements

In this paper, three different machine learning classifiers were trained and tested on a dataset created from the combination of RAVDESS[7] and CREMA[3] datasets. As it was predictable, the KNN obtained the worse result since it is the simplest classifier. Considering the speed of training process it was not a bad result. SVM obtained a very similar result in comparison with CNN classifier and its training time was definitively shorter than the deep learning method.

In future works, the results obtained can be improved by selecting and combining other time and frequency domain features in addition to the ones used. It can be also interesting evaluate other machine learning classifiers to see if they can perform better than the ones used in this project.

REFERENCES

[1] Sharmeen M Saleem Abdullah Abdullah et al. "Multimodal emotion recognition using deep learning". In: *Journal of Applied Science and Technology Trends* 2.02 (2021), pp. 52–58.

[2] Tanja Bänziger, Didier Grandjean, and Klaus R Scherer. "Emotion recognition from expressions in face, voice, and body: the Multimodal Emotion Recognition Test (MERT)." In: *Emotion* 9.5 (2009), p. 691.

[3] Houwei Cao et al. "Crema-d: Crowd-sourced emotional multimodal actors dataset". In: *IEEE transactions on affective computing* 5.4 (2014), pp. 377–390.

[4] Rajdeep Chatterjee et al. "Real-time speech emotion analysis for smart home assistants". In: *IEEE Transactions on Consumer Electronics* 67.1 (2021), pp. 68–76.

[5] Marwan Dhuheir et al. "Emotion recognition for healthcare surveillance systems using neural networks: A survey". In: *2021 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE. 2021, pp. 681–687.

[6] Paul Gavrikov. *visualkeras*. https : / / github . com / paulgavrikov/visualkeras. 2020.

[7] Steven R Livingstone and Frank 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". In: *PloS one* 13.5 (2018), e0196391.

[8] Brian McFee et al. "librosa: Audio and music signal analysis in python". In: *Proceedings of the 14th python in science conference*. Vol. 8. 2015, pp. 18–25.

[9] Nicu Sebe, Ira Cohen, and Thomas S Huang. "Multimodal emotion recognition". In: *Handbook of pattern recognition and computer vision*. World Scientific, 2005, pp. 387–409.

[10] Taiba Majid Wani et al. "A comprehensive review of speech emotion recognition systems". In: *IEEE access* 9 (2021), pp. 47795–47814.

[11] Teng Zhang and Ji Wu. "Speech emotion recognition with i-vector feature and RNN model". In: *2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*. IEEE. 2015, pp. 524–528.