Audio Pattern Recognition Project: Speech Emotion Recognition

Matteo Onger

April 2024

1 Introduction

This project aims to implement a model for solving Speech Emotion Recognition tasks, that is, detecting the speaker's emotional state from the speech signal, based on the way he/she spoke and regardless of what he/she said. A growing interest in this type of tasks has developed due to various practical applications [3]: from the medical/psychiatric field to intelligent toys, from call center conversations to car board system, where information about the mental state of the driver may be provided to the system to ensure people's safety. The ambition of the project is to build a simple neural network able to predict emotions with sufficient accuracy and to evaluate how different hyper-parameters may affect the results.

The document is structured as follows: the second and the third chapters present the dataset and the approach followed, while the fourth one presents the results of the tests conducted before reaching the final conclusions.

2 Dataset Description

The dataset used is known as CREMA-D (Crowd-sourced Emotional Multimodal Actors Dataset) [1] and it contains 7.442 audio files from 91 actors (48 male and 43 female actors) repeating one among twelve possible sentences of a few seconds (1-5 seconds). The emotions considered are the following:

1. Anger (ANG);
2. Disgust (DIS);
3. Fear (FEA);
4. Happy (HAP);
5. Neutral (NEU);
6. Sad (SAD).

The dataset is substantially balanced, not only from the perspective of the actors' gender, but also in terms of emotions. The figure 1 below shows the distribution of the data.

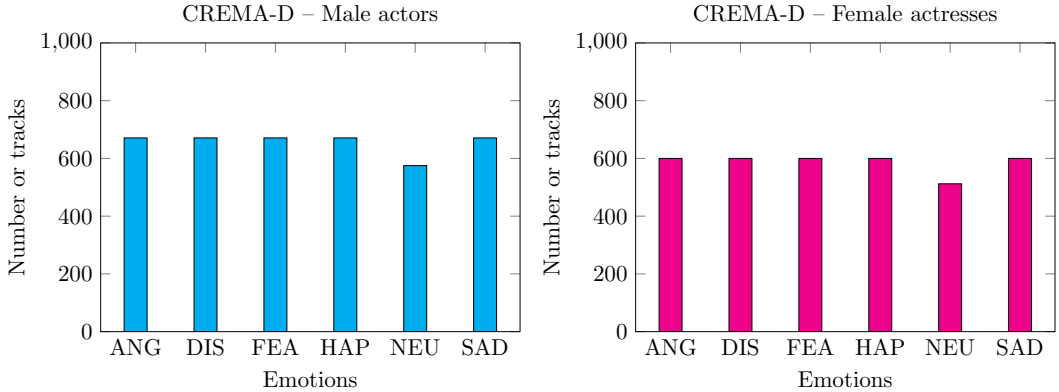

Figure 1: Number of audio tracks per emotion.

3 Model Description

The steps performed for training, and later for testing, are shown in figure 2. During the training phase, the first 3.5 seconds of each audio track are loaded, skipping the first 0.5 seconds, which are considered less significant. In order to increase the size of the dataset, some standard data augmentation techniques are applied to the original audio tracks: adding a controlled but random quantity of noise, shifting the pitch and combing both these methods allows us to quadruple the number of data available.

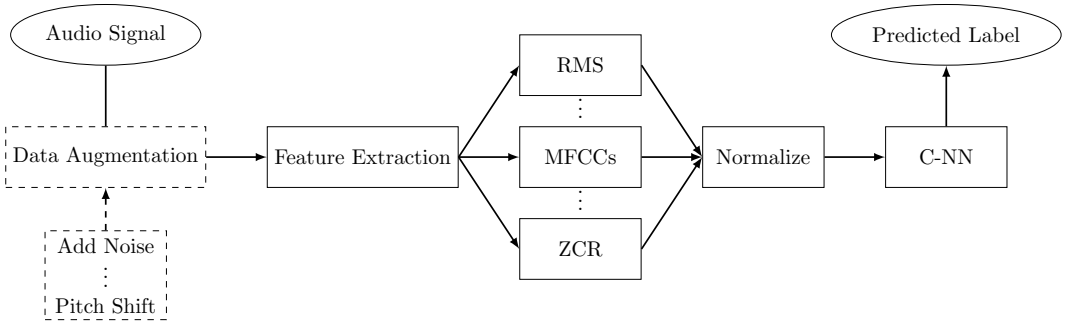

Figure 2: Diagram of the model. Dashed steps are performed only during training.

For each audio signal a set of features is computed. Obviously, the greater the number of features calculated, the greater the probability that at least some of them will actually be useful for the purpose of classifying the signal, but this also implies a greater computational complexity. So it would be better, as far as possible, to keep only the features that are most significant for the specific problem being addressed. In this project, the following features were used:

- Mel-frequency cepstral coefficients (MFCCs);
- Root-mean-square value: $RMS(frame) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}|x_i|^2} \quad \forall x_i \in frame$;
- Zero-crossing rate: $ZCR(frame) = \frac{1}{2n}\sum_{i=2}^{n}|sign(x_i) - sign(x_{i-1})| \quad \forall x_i \in frame.$

For more details about these features, please refer to [4].
All these features, for each audio tracks, are saved in a one-dimensional array: k Mel coefficients are saved for each of the n frames, thus forming a vector of length kn. To it, n values representing the RMS of each frame and other n values representing the ZCR of each frame are appended. So the final length of the vector is equal to $kn + 2n$.
Obviously, not only the features but also their parameters, such as the overlap percentage or the frame length, can significantly affect the results and are crucial to balancing temporal and frequency resolution.

At this point, each feature is standardized using the formula 1, where u is the mean and s is the standard deviation of the training samples.

$$z = \frac{x - u}{s} \tag{1}$$

Finally, the normalized features are given as an input to the one-dimensional convolutional neural network, the output of which is the predicted label. During the training procedure, particularly during the back-propagation phase, the predicted label is compared with the expected label; the error committed is then used to update the network parameters using one of the variants of the gradient descent.

Neural networks differing in their structure and/or in their hyper-parameters were compared, as well as different parameters for feature generation were tested; the results collected are presented in the following chapter.

4 Experimental Results

4.1 CNNs and hyper-parameters

The neural network originally taken as a benchmark is illustrated in figure 3 and more information can be found in document [2].

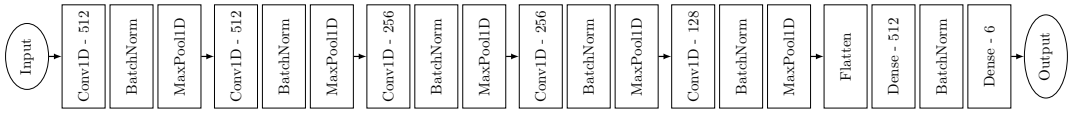

Figure 3: Structure of the 1D-CNN taken as a benchmark. The numbers shown in the figure indicate the filters per level, while $kernel\_size = 5$ and $strides = 1$.

This neural network has been shown to predict emotions given an audio file with excellent accuracy: above 90% after 25 epochs and a batch size of 64. But it is a CNN of non-negligible size, especially due to the number of filters used in the various layers. Therefore, an attempt was made to construct a "simpler" neural network that could provide equivalent performance.

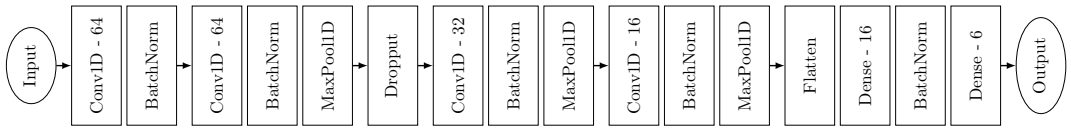

Figure 4: Structure of the proposed 1D-CNN. The numbers shown in the figure indicate the filters per level.

The figure 4 shows the one-dimensional convolutional neural network proposed: it has 16 layers, each one with a significant smaller number of filters, and, in particular, the sixth is a dropout layer with $rate = 0.2$. But one of the most important aspects is the choice of two hyper-parameters: the kernel size and the strides. The following table shows the accuracy trend as these hyper-parameters change.

N°	Kernel Size	Strides	Accuracy (%)
1°	5	1	72.86
2°	20	1	83.64
3°	20	20	90.86
4°	40	1	87.68
5°	40	20	93.95
6°	40	40	92.12

Table 1: Accuracy on the test set of the proposed neural network as kernel size and strides change.

During these tests, 20 Mel coefficients were always computed for each of the 110 frames; this may explain why setting the hyper-parameters to multiples of 20 yields better results. The graph 5 shows the accuracy achieved by the neural networks during the epochs, using a batch size of 64.

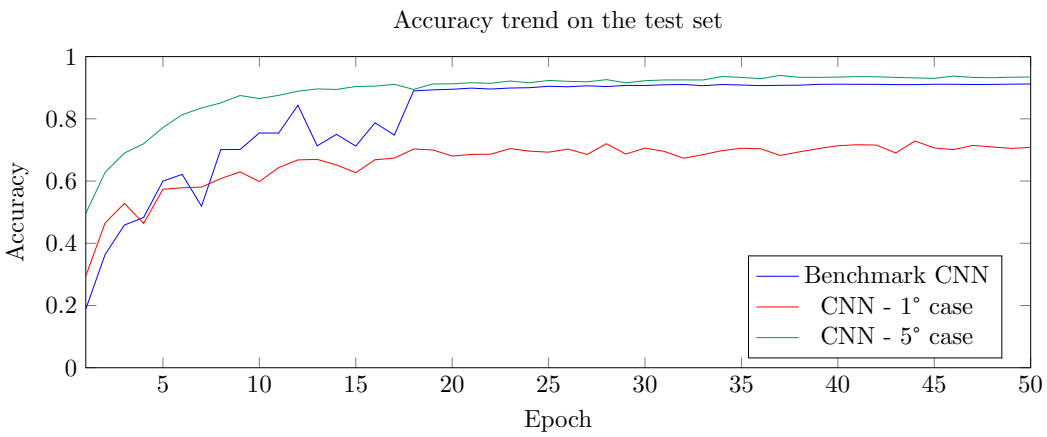

Figure 5: Test accuracy of the three CNNs: the one taken as a benchmark, the one proposed with $kernel\_size = 5$ and $strides = 1$ or with $kernel\_size = 40$ and $strides = 20$.

Lastly, it should be noted that by using the softmax function as the last layer, the output produced is actually not one of the possible labels, but a vector with an entry for each emotion: the i-th element is the probability calculated by the network that the label associated with the input provided is the i-th emotion. Obviously the predicted emotion remains the one with the highest probability, but in this way it is possible to know the degree of confidence with which the network is returning a certain prediction.

ANG	DIS	FEA	HAP	NEU	SAD
0.92	0.05	0.03	0.0	0.0	0.0

Figure 6: A possible example of a vector returned by the CNN as output.

4.2 Features

Even higher accuracy can be achieved by exploiting an additional information provided by the CREMA-D dataset: the gender of the speaker. Two identical neural networks were trained: one using only audio tracks recorded by male actors and the other one using only audio tracks from female actresses. Despite the smaller amount of data available to train each of the two networks, the performance obtained was better due to the greater uniformity. This is shown in the graph 7.

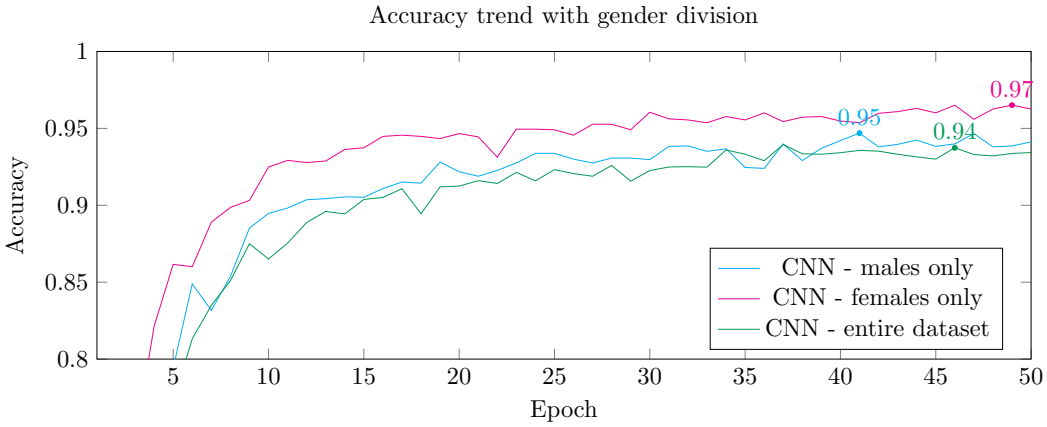

Figure 7: Accuracy of the proposed 1D-CNN with $kernel\_size = 40$ and $strides = 20$ trained on the three datasets: the full one, the one containing only the audios of male speakers, and the one containing only the audios of female speakers.

A final set of experiments was done in order to test the impact that feature's parameters have on the accuracy of the neural network. The parameters considered were: the frame length (measured in number of samples per frame), the overlap percentage and the number of Mel-frequency cepstral coefficients computed per frame. Table 2 shows the combinations tried.

N°	Frame Length	Overlap Percentage	#MFCCs	Max Accuracy (%)
1°	4096	25%	20	94.14
2°	2048	25%	20	94.85
3°	1024	25%	20	91.41
4°	512	25%	20	85.68
5°	2048	25%	10	91.86
6°	2048	25%	30	94.78
7°	2048	0%	20	87.25
8°	2048	50%	20	91.83

Table 2: Combinations of the tested features' parameters and the resulting maximum accuracy achieved by the proposed neural network on the test set.

Unfortunately, as also clearly shown by charts 8, 9 and 10, none of the combinations tried led to an improvement in performance; on the contrary, some of them led to significantly worse accuracy. Thus, the collected data seem to show that the proposed neural network reaches its maximum accuracy when used with the following parameters: $kernel\_size = 40$, $strides = 20$, $frame\_length = 2048$, $overlap\_percentage = 25\%$ and $\#MFCCs = 20$.

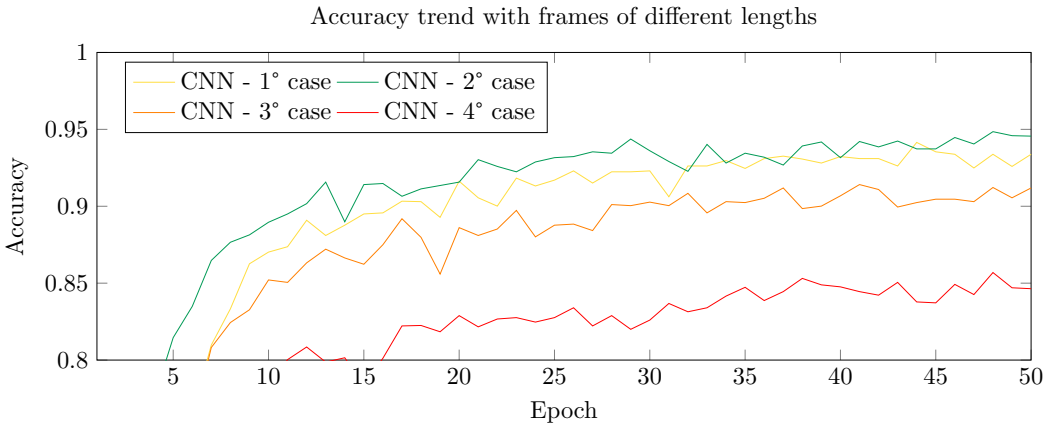

Figure 8: Accuracy on the test set of the proposed neural network as frame size changes.

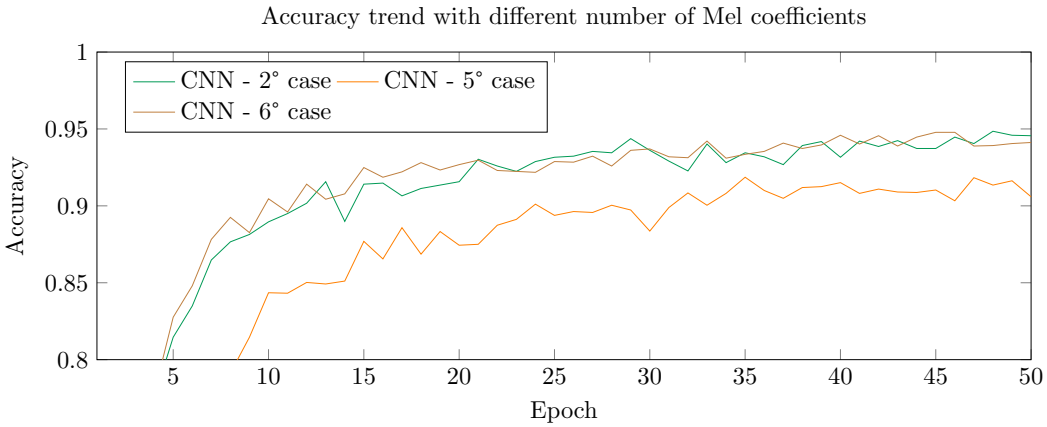

Figure 9: Accuracy on the test set of the proposed neural network as the number of Mel-frequency cepstral coefficients changes.

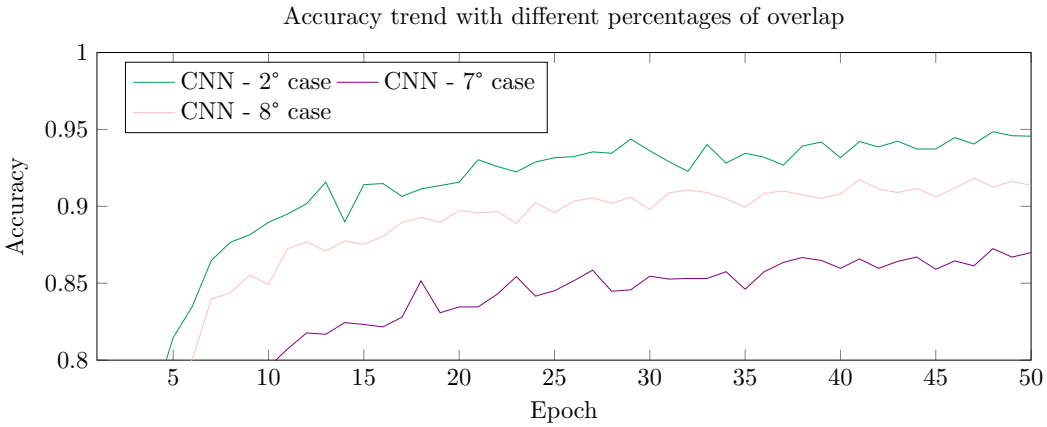

Figure 10: Accuracy on the test set of the proposed neural network as the overlap percentage changes.

5 Conclusions

In conclusion, the proposed neural network, using appropriate hyper-parameters, was shown to be able to classify emotions with good accuracy, as also shown by the confusion matrices 11. The few mistakes are often made by confusing emotions that are similar to each other, or emotions that are harder to define exactly, such as anger and disgust. And all this using an overall fairly small model.

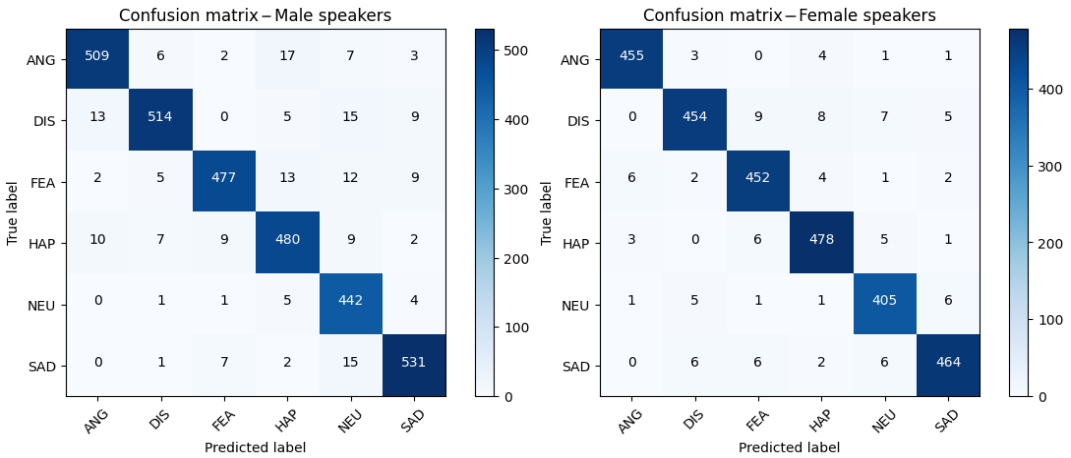

Figure 11: Confusion matrix for audio tracks of male and female speakers.

Of course, many other studies or improvements could be done, starting from the code to extract the features that could be parallelized in order to reduce the computation time. And many other tests regarding the role of hyper-parameters and features could be done in a more systematic way, using specific tools such as KerasTuner. The structure of the proposed neural network could be changed, or completely different neural networks (such as 2D-CNNs, RNNs, LSTMs, etc.) and model not based on NNs could be tested. In any case, the proposed network can be seen as an interesting starting point.

References

[1] 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.

[2] Speech emotion recognition (conv1d) - kaggle. https://www.kaggle.com/code/dmitrybabko/speech-emotion-recognition-conv1d.

[3] Ashish B Ingale and DS Chaudhari. Speech emotion recognition. *International Journal of Soft Computing and Engineering (IJSCE)*, 2(1):235–238, 2012.

[4] Librosa: a python package for music and audio analysis. https://librosa.org/doc/0.10.1/index.html.