# Classification of Emotions using EMO-DB

Chiara Genuardi, Giovanni Noè

**Abstract**

This project investigates emotion recognition using the Berlin Database of Emotional Speech (EMO-DB). It contains audio recorded in a controlled environment by 10 professional actors and actresses, expressing seven different emotions in established sentences.

After applying preprocessing steps and Mel-frequency Cepstral Coefficient (MFCC) extraction we trained several deep learning models applying the LOSO cross validation technique and data augmentation on training dataset. The evaluation showed that the best performances are obtained with the combination of 2D convolutional layers and LSTMs.

We then assessed generalization by deploying the model using audio files recorded by non-professionals; the resulting performance drop highlights the impact of domain shift and underlines the importance of dataset diversity and robustness in emotion recognition research.

## Table of Contents

**Introduction** ........ 1

**1. Data set Description and Brief Exploration** ........ 2

**2. Pre-processing** ........ 3

Resampling ........ 3

Filtering ........ 3

Conversion of Input Shape ........ 3

**3. Data Exploration and Feature Extraction** ........ 3

Wav Exploration ........ 3

MFCC Exploration ........ 4

**4. LOSO Cross-Validation** ........ 4

**5. Data Augmentation** ........ 5

**6. Models** ........ 5

Multi Layer Perceptron ........ 6

Convolutional Neural Network 2D ........ 7

Convolutional Neural Network 2D + Long Short Term Memory ........ 8

Overall conclusions ........ 8

**7. Test Phase** ........ 9

**Conclusions** ........ 10

**References** ........ 10

## Introduction

Emotion recognition is a rapidly growing research area with several applications; the ability to infer automatically emotional states would make virtual assistants more natural and improve automated customer service systems.

Generalization remains challenging, due to variability between speakers, differences in recording conditions and the difference between acted and spontaneous emotions; all those factors make it crucial to explore different preprocessing strategies.

Our project followed a typical pipeline.

To begin we pre-processed wav data at our disposal to enhance its quality and prepare it for further analysis, by resampling, filtering and converting their input shape.

Data preparation ends with feature extraction, critical in audio deep learning models; we used Mel-frequency Cepstral Coefficients (MFCC), a way to represent the short-term power spectrum of a sound which helps machines understand and process human speech effectively.

Since we are considering 10 people, 10 sentences and 7 different emotions, to avoid models learnt how different people expressed specific emotions we applied a cross-validation technique, and particularly the leave-one-speaker-out (LOSO), such that the training consists of 10 cycles, each done on 9 speakers while the last one is used to validate the model.

As said the database consists of a total of 535 audio files; to increase the size and diversity of training data we augmented them for each LOSO cycle by adding noise and shifting audio, as it helps to improve generalization.

The core of the project consists in training different models, such as a multi-layer perceptron, convolutional neural networks, a recurrent neural network and combinations of them.
Their evaluation considering different metrics showed that the best performing model was a combination of CNN2D and LSTM.

At last, we tested the model on audio files recorded by 4 non-professionals, showing the importance of domain shift.

## 1. Data set Description and Brief Exploration

The Berlin database of Emotional Speech is a collection of sound files, label files and perception test results of emotional utterances spoken in German by actors in an anechoic chamber, recorded in 1997 and 1999 in the University of Berlin as part of the DFG funded research project.
The database contains a total of 535 utterances pronounced by 5 actors and 5 actresses which simulate different emotions (anger, boredom, disgust, fear/anxiety, happiness, sadness, neutral version) in 10 different sentences.

We are going to use the wav folder which contains all the audio files, which are named like this:

- Position 1-2: numbers that identify the speaker
    - 03 – male, 31 years
    - 08 – female, 34 years
    - 09 – female, 21 years
    - 10 – male, 32 years
    - 11 – male, 26 years
    - 12 – male, 30 years
    - 13 – female, 32 years
    - 14 – female, 35 years
    - 15 – male, 25 years
    - 16 – female, 31 years
- Position 3-5: one letter and two numbers that identify the sentence, there’s a total of 10 sentences spoken in German
- Position 6: letter that identifies the emotion
    - W – Arger (Wut), anger
    - L – Langeweile, boredom
    - E – Ekel, disgust
    - A – Angst, anxiety/fear
    - F – Freude, happiness
    - T – Trauer, sadness
    - N – Neutral
- Position 7: if there are more versions these are numbered with letters a, b, c and so on.

In the database there isn’t the same number of recordings for each speaker, and neither emotions are expressed in the same number of sentences; in particular, for speakers we have:

- Speaker 03: 49 files
- Speaker 08: 58
- Speaker 09: 43
- Speaker 10: 38
- Speaker 11: 55
- Speaker 12: 35
- Speaker 13: 61
- Speaker 14: 69
- Speaker 15: 56
- Speaker 16: 71

and for emotions:

- Disgust: 46 sentences
- Sadness: 62
- Happiness: 71
- Neutral: 79
- Anger: 127
- Fear: 69
- Boredom: 81.

Audio’s lengths span between 1.23 seconds to 8.98, but most of them (401) have lengths between 1.5 and 3.5 seconds.

## 2. Pre-processing

Pre-processing is a crucial step in the pipeline of audio data deep learning and includes different techniques applied to raw data to enhance its quality, extract meaningful features and prepare it for further use and can significantly impact the performance of models trained on them.

Three important concepts around signals are sample rate, frequency of the signal and input shape: we're working on them in this pre-processing phase.

### Resampling

Sample rate, measured in hertz, defines the number of samples or signal measurements which are registered per second, the higher the signal the more detailed the representation of the frequency spectrum.

Resampling is a technique about standardizing the sample rate of audio signals, it helps in mitigating issues related to mismatched sample rates and enhancing computational efficiency of subsequent processing steps.

The sample rate chosen is 16000Hz, which is the used standard; if the audio signal has a different sample rate is then resampled.

### Filtering

Frequency of the signal, measured in hertz, indicates how many oscillations per second a sound wave makes. Filtering is used to modify the frequency content of an audio signal by attenuating or emphasizing certain frequency components.

It is applied a low pass filtering to remove high frequency noise which ensures that the model focuses on relevant signal information; in this case is used as a threshold a frequency of 4000Hz, chosen since most of the information useful to understand the spoken word is under this value, as human voices span between 80 to 400Hz, reaching the order of kHz. All noise over this value is attenuated.

This way noise is reduced, and the model can learn on the actual information we need, which is the person saying the specific sentence and not background noise.

### Conversion of Input Shape

This last step involves shaping the raw audio signal, deciding which is the number of samples maintained. This step includes operations such as trimming and padding, ensuring the model can effectively process and learn from the audio data during training or make accurate predictions during inference.

In this case we decided to maintain a fixed dimension of 64000 samples for each signal, keeping the central section. The reason behind this choice lies in the fact that generally the most representative information is stored in the middle, so this avoids the loss of the important parts.

The target length of 64000 corresponds circa to 4 seconds, this choice was made since the audio in the database span between 1.2 and 8.9 seconds, this way we're keeping the average length as target.

If the audio is too short then padding is operated, which involves adding zeroes both at the left and at the right of signal, keeping the original audio in the middle; if it is too long then the signal is trimmed maintaining the central part of the audio of the targeted length.

## 3. Data Exploration and Feature Extraction

### Wav Exploration

Before extracting features to be used to train the models we performed basic analysis on the processed wav files.

We compared two files spoken by the same actor of the same sentence, the first in the neutral version and the second expressing anger to highlight noticeable acoustic differences.

Amplitude reflects the intensity of an audio signal at a particular point of time; the amplitude range of a signal provides information about its dynamic range. A waveform can be visualized by showing how its amplitude varies over time, highlighting pauses and general energy patterns of speech (Figure 1).

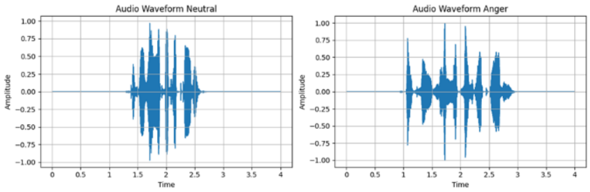


*Figure 1 - Audio Waveform Neutral vs Anger*

The neutral version has a shorter duration, while the angry version takes slightly longer, as if the person took more care on spelling out words well; the neutral version shows how the amplitude grows smoothly and then decays, while the angry one has a less soft contour, having sharper amplitude variations.

A frequency spectrum represents how energy in an audio signal is distributed across different frequencies (Figure 2).

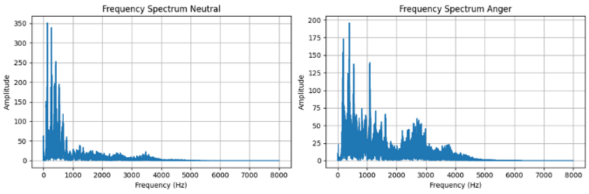


*Figure 2 - Frequency Spectrum Neutral vs Anger*

The neutral version shows small values in amplitude, mostly above 1000Hz, while the angry version keeps discretely high values extending up to around 3000Hz.

Since deep learning models rarely use raw audio as a direct input, signals are generally converted into spectrograms (Figure 3). A spectrogram is a time-frequency representation of an audio signal generated applying Short Time Fourier Transform, providing a two-dimensional visualization of how the frequency content of the audio signal changes over time.

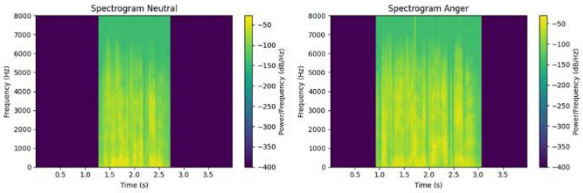


*Figure 3 - Spectrogram Neutral vs Anger*

In our comparison the angry version reaches higher values of power across frequency.

**MFCC Extraction**

Mel-Frequency Cepstral Coefficients (MFCC) are the most widely used features in speech and audio recognition as they capture the salient features of speech; MFCCs enable computers to discern between distinct words and sounds.

When computed over time they form a two-dimensional representation that resembles an image; for this reason, this makes them suitable for Convolutional Neural Networks (CNNs). At the same time MFCCs are inherently sequential, so they can be processed with Recurrent Neural Networks (RNNs) such as Long Short Time Memory (LSTMs) to capture model dependencies, while multi-layer perceptrons (MLPs) can exploit them as compact feature vectors for classification.

The extraction happens by dividing the waveform into short overlapping frames and applying a Fourier transform to obtain the frequency spectrum. This is filtered with a Mel-scale filterbank, which approximates the non-linear frequency resolution of the human auditory system by emphasizing low frequencies, where most speech information lies. A logarithmic compression is applied to simulate the human perception of loudness, followed by a Discrete Cosine Transform (DCT) to decorrelate the coefficients and retain only the most informative components. We extracted 13 coefficients per frame (a standard choice in speech tasks), using an FFT size of 1024 and a hop length of 512. These parameters strike a balance between temporal resolution and frequency resolution: a 1024-point FFT at 16 kHz sampling rate provides sufficient detail to capture the phonetic structure of speech, while a hop length of 512 ensures a reasonable overlap between frames without excessive redundancy. This configuration allows the model to capture both the fine-grained spectral patterns and the broader temporal dynamics of emotional prosody.

Feature extraction was performed with the librosa.feature.mfcc function.
Figure 4 shows the MFCC representation on the neutral sentence illustrated above.

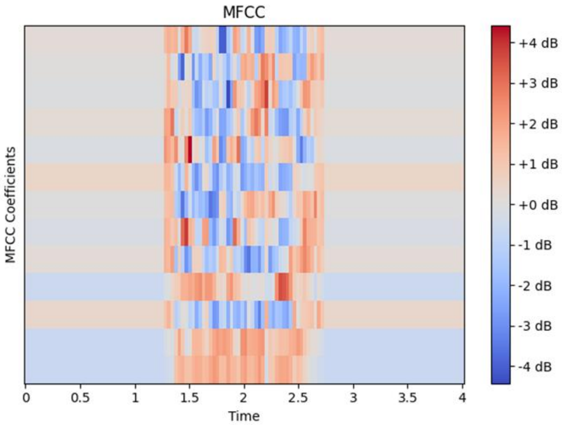


*Figure 4 - MFCC Neutral Version*

The temporal evolution of coefficients is displayed, as each vertical stripe corresponds to a short time window, and the intensity of colours reflect how the spectral content varies. Lower-order coefficients capture global spectral properties such as overall energy while higher-order's encode finer details.

## 4. LOSO Cross-Validation

One of the most critical aspects of our project is the implementation of LOSO cross-validation. LOSO (Leave-One-Speaker-Out) is a variant of K-fold cross-validation in which each speaker's audio recordings are treated as a separate fold. Specifically, for each model, ten training–validation cycles are performed, one for each speaker. In every cycle, the training set consists of the recordings from nine speakers, while the validation set contains only the recordings of the held-out speaker.

This process is employed during model development to improve the reliability of performance estimates, providing a measure of how well the models generalize to previously unseen data originating from speakers who are not included in the training set.

Since the LOSO procedure is identical for all models, the dataset is partitioned into ten folds only once. We first define a function that takes the preprocessed data and the speaker ID as input and generates a new directory containing the LOSO split. The split is organized into training data (nine speakers) and test data (the held-out speaker), with each subset further divided into seven subfolders corresponding to the emotion categories.

We then implement the LOSO cycle by applying this function to each speaker, storing the resulting data as TensorFlow datasets. Data augmentation is applied exclusively to the training sets, while the validation sets remain unchanged (see the next section for details on data augmentation). Then, the training set is further processed by batching and shuffling, whereas the validation set is created in a batched format without shuffling. Once the cycle is complete, the data are ready to be used for model training and validation.

Lastly, it is important to emphasize that, although LOSO cross-validation helps reduce dependence on the training data, EMO-DB is still a dataset collected in a highly controlled and artificial environment. Consequently, further evaluation on more diverse and realistic data sources is required to obtain a reliable assessment of the model's quality and generalizability.

## 5. Data Augmentation

With only 535 recordings in the EMO-DB dataset, all collected in a highly controlled environment, the issue of generalization is particularly critical for this project. While LOSO cross-validation was employed to improve the reliability of performance evaluation, enhancing the model's ability to generalize requires the use of data augmentation techniques.

In this work, we adopt two augmentation strategies for audio files:

- Noise addition: a random vector with values between 0 and 0.005 is added to each preprocessed audio sample in its MFCC representation.
- Time shifting: the audio signal is shifted forward or backward by up to 0.8 seconds.

Since EMO-DB also suffers from class imbalance, we addressed this issue within the data augmentation process. Specifically, during the LOSO cycle, for each speaker's training dataset, we iterated over the samples and applied augmentation techniques only when the emotion associated with the current sample had fewer than 150 recordings in the augmented dataset at that point. The threshold of 150 was chosen as it corresponds to approximately three times the number of samples associated with the least represented emotion in the dataset. Moreover, data augmentation was integrated into the LOSO cycle to ensure that each model was trained on the same augmented dataset.

Finally, through this data augmentation strategy, we achieved balanced classes within each training set and obtained augmented training datasets approximately two to three times larger than the original ones, with the exact sizes depending on the number of recordings available for each speaker.

## 6. Models

In this project, we explored a wide range of deep learning architectures to identify the one best suited for our audio classification task:

- a Multilayer Perceptron (MLP), used as a baseline;
- two Convolutional Neural Networks (a 1D CNN and a 2D CNN);
- a Long Short-Term Memory (LSTM) Recurrent Neural Network;
- a hybrid model combining an LSTM with a 1D CNN;
- a hybrid model combining an LSTM with a 2D CNN.

All these models are trained using a RMSprop optimizer (except for the MLP that employs Adam) with a learning rate of 0.001 for up to 25 epochs with early stopping based on validation accuracy.

To evaluate the tested models, we employ multiple performance measures: accuracy, top 3 accuracy, precision, recall, and the F1 score. In addition, we analysed the confusion matrix, ROC curves, and AUC values.

Since LOSO cross-validation was implemented, accuracy and top 3 accuracy are computed separately for each speaker and then averaged. The remaining metrics, along with a non-aggregated version of accuracy, are calculated using the combined predictions from all speakers.

Each model in this project is evaluated under multiple configurations to identify the best-performing variant. However, since most models (i.e., CNN1D, LSTM, and LSTM+CNN1D) achieved results comparable to the MLP baseline, this report discusses only the baseline and the two best-performing models.

**Multi Layer Perceptron**

The first architecture we implemented was a standard Multilayer Perceptron (MLP), used as a baseline for comparison with the other models.

Since MLPs operate on one-dimensional inputs, we reduced the dimensionality of the MFCC-shaped audio recordings (originally of shape (126, 13)). To do so, we aggregated the time dimension by computing the mean and standard deviation for each of the 13 MFCC features. The resulting input vector for the MLP has a length of 26: 13 features represent the temporal averages of the MFCC coefficients, while the remaining 13 represent their corresponding standard deviations over time.

The architecture consists of five dense layers, with batch normalization applied between each pair of layers. The first four layers use the ReLU activation function, with 256, 128, 64, and 32 neurons respectively. The final layer maps the output to the classification space, consisting of 7 neurons, one for each emotion class.

The average accuracy across individual speakers in LOSO cross-validation was 0.50, with a top 3 accuracy of 0.85. However, when considering all predictions collectively, the overall accuracy increased to 0.64.

Classification metrics from the classification_report function (Table 1), confusion matrix (Figure 5) and ROC (Figure 6) are here reported:

| | precision | recall | F1-score | Support |
|---|---|---|---|---|
| Anger | 0.81 | 0.83 | 0.82 | 127 |
| Boredom | 0.63 | 0.27 | 0.38 | 81 |
| Disgust | 0.50 | 0.70 | 0.58 | 46 |
| Fear | 0.75 | 0.48 | 0.58 | 69 |
| Happiness | 0.49 | 0.56 | 0.52 | 71 |
| Sadness | 0.75 | 0.74 | 0.75 | 62 |
| Neutral | 0.52 | 0.77 | 0.62 | 79 |
| | | | | |
| Accuracy | | | 0.64 | 535 |
| Macro avg | 0.64 | 0.62 | 0.61 | 535 |
| Weighted avg | 0.66 | 0.64 | 0.63 | 535 |

*Table 1 - Metrics MLP*

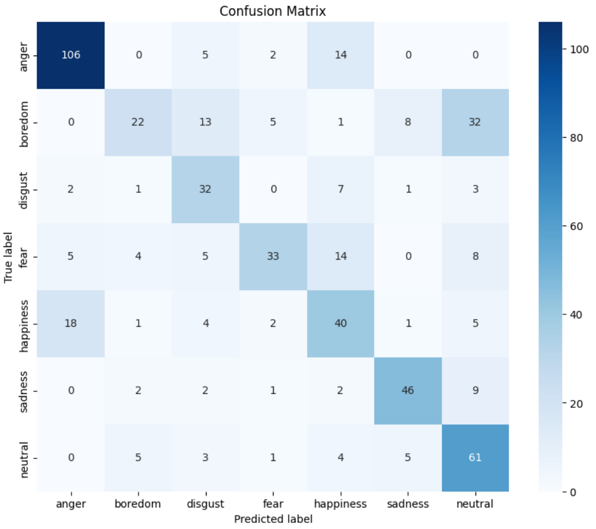


*Figure 5 - Confusion Matrix MLP*

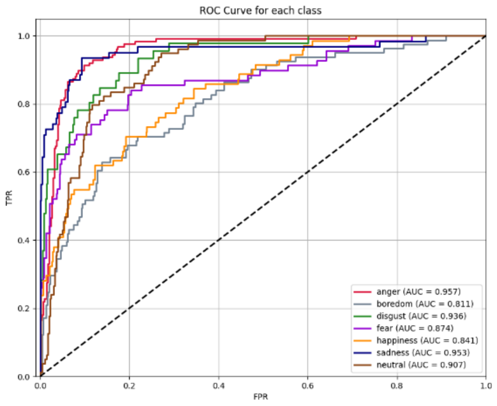


*Figure 6 - ROC curve MLP*

Analysis of the ROC curves shows that AUC values range from 0.811 for boredom to 0.957 for anger.

Overall, the model achieves a solid performance, with a weighted average F1-score of 0.63.
The best-recognized emotions are anger and sadness, obtaining very similar AUC values. In contrast, boredom and happiness are the most frequently misclassified emotions, often confused with neutral and anger respectively, fact that is clearly reflected in the evaluation metrics.

Therefore, this basic MLP achieves reasonably good results and serves as the baseline for all subsequent models.

### Convolutional Neural Network 2D

The second best-performing model in our experiments is a 2D Convolutional Neural Network (CNN2D).

The input has a shape of (126, 13), representing sequences of 126 frames with 13 MFCC coefficients each. The network consists of four convolutional blocks with kernel size 5 and filter sizes increasing from 16 to 128. Each block is followed by ReLU activation and max pooling, with batch normalization applied in the first block.

After the convolutional layers, global max pooling condenses the feature maps into a fixed-length vector, which is then passed through a dense layer with 64 units and ReLU activation, before the final softmax output layer that predicts class probabilities.

This architecture achieves an average accuracy of 0.63 and a top 3 accuracy of 0.93, with an overall accuracy of 0.71.

Classification metrics are here reported (Table 2, Figure 7, 8):

| | precision | recall | F1-score | Support |
|---|---|---|---|---|
| Anger | 0.77 | 0.92 | 0.84 | 127 |
| Boredom | 0.83 | 0.56 | 0.67 | 81 |
| Disgust | 0.63 | 0.57 | 0.60 | 46 |
| Fear | 0.60 | 0.61 | 0.60 | 69 |
| Happiness | 0.60 | 0.44 | 0.50 | 71 |
| Sadness | 0.90 | 0.90 | 0.90 | 62 |
| Neutral | 0.62 | 0.82 | 0.71 | 79 |
| | | | | |
| Accuracy | | | 0.71 | 535 |
| Macro avg | 0.71 | 0.69 | 0.69 | 535 |
| Weighted avg | 0.72 | 0.71 | 0.71 | 535 |

*Table 2 - Metrics CNN2D*

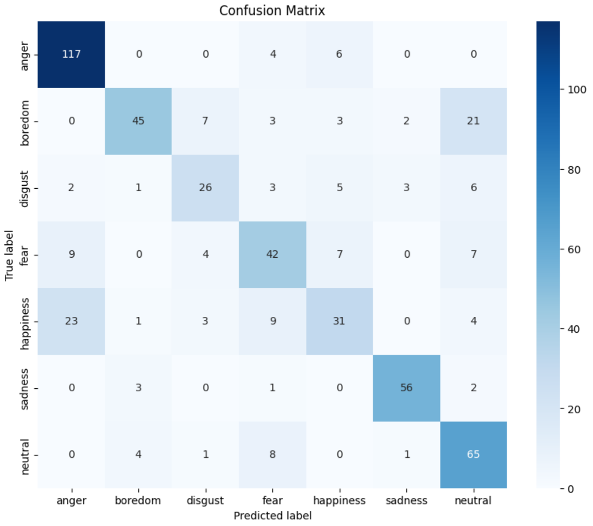


*Figure 7 - Confusion Matrix CNN2D*

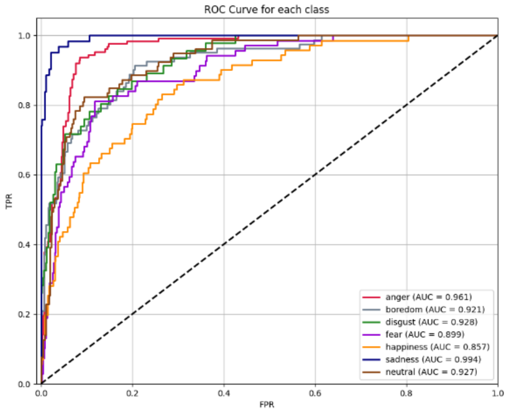


*Figure 8 - ROC curve CNN2D*

From the ROC curve we observe that sadness reaches the highest AUC value (0.994) while happiness records the lowest (0.857).

Overall, the CNN2D performs better than the previous architectures. Sadness is the best-classified emotion, while happiness seems to be the most difficult to distinguish.

This model proves to be a robust option for emotion classification, although certain samples are still misclassified.

### Convolutional Neural Network 2D + Long Short Term Memory

The best tested network is a hybrid architecture that mixes CNN2D and LSTM.
The input has shape (126, 13, 1), corresponding to sequences of 126 frames with 13 MFCC coefficients each, treated as a single-channel 2D feature map. The architecture is composed of two convolutional blocks, where convolutional layers use a ReLU activation and are followed by a batch normalization and a max pooling 2D layer.
The output of the convolutional part is reshaped into a temporal sequence to be processed by recurrent layers. Two LSTMs layers are applied: the first with 128 units, and the second with 64 to capture higher-level sequential patterns. At last, a final softmax dense output layer predicts class probabilities for the seven emotions classified.

This architecture achieves an average accuracy of 0.65 and a top 3 accuracy of 0.95, with an overall accuracy of 0.76.

Classification metrics are here reported (Table 3, Figure 9, 10):

| | precision | recall | F1-score | Support |
|---|---|---|---|---|
| Anger | 0.80 | 0.96 | 0.87 | 127 |
| Boredom | 0.76 | 0.59 | 0.67 | 81 |
| Disgust | 0.79 | 0.74 | 0.76 | 46 |
| Fear | 0.80 | 0.71 | 0.75 | 69 |
| Happiness | 0.64 | 0.51 | 0.57 | 71 |
| Sadness | 0.89 | 0.22 | 0.86 | 62 |
| Neutral | 0.65 | 0.85 | 0.74 | 79 |
| | | | | |
| Accuracy | | | 0.76 | 535 |
| Macro avg | 0.76 | 0.74 | 0.75 | 535 |
| Weighted avg | 0.76 | 0.76 | 0.75 | 535 |

*Table 3 - Metrics CNN2D+LSTM*

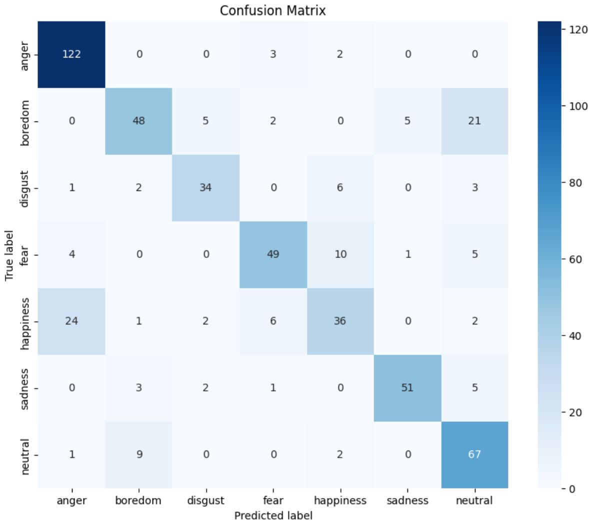


*Figure 9 - Confusion Matrix CNN2D+LSTM*

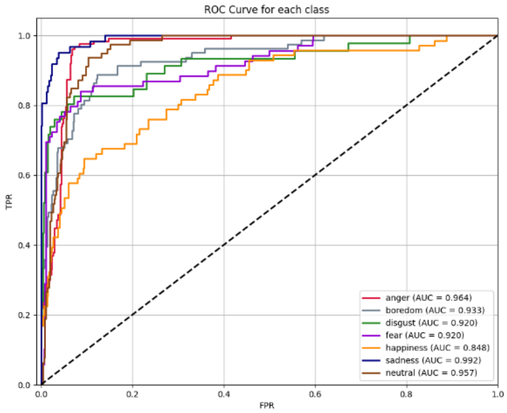


*Figure 10 - ROC CNN2D+LSTM*

The AUC values range from 0.848 for happiness to 0.992 for sadness.
This model delivers overall better performance compared to the previous ones.
Happiness is still the most challenging, obtaining a recall of 0.51, as it is often confused for anger.

Compared to earlier models, this architecture shows a stronger ability to correctly classify most classes, with errors confined to specific pair of classes.

This model demonstrates greater robustness than the previous and is selected for further testing.

### Overall conclusions

Although the different architectures exhibited specific strengths and weaknesses, some consistent patterns emerged across all of them.

Sadness was the most reliably recognized emotion, achieving consistently high F1-scores and the highest AUC values overall. Anger also achieved strong results, benefiting from the relatively high number of samples available in the dataset.

By contrast, happiness was the most challenging emotion to detect, with low values both for F1-scores and AUCs; it was frequently misclassified as anger or, at times, fear.

Another recurring issue was the confusion between boredom and neutral, which were often misclassified with each other and, in some cases, with disgust.

## 7. Test Phase

As discussed in the Data Augmentation section, a major concern of our project was the generalization ability of the models. While LOSO cross-validation provided a more realistic estimate of performance, it was still important to further challenge the models with entirely new data recorded in less controlled environments and with non-professional microphones. To this end, we attempted to replicate the procedure followed by the researchers at the University of Berlin by recording audio samples of ourselves and acquaintances simulating emotions, which were then used as input for classification by our model. Since our goal was to make the model as data-independent as possible, and the EMO-DB dataset did not provide enough samples to create a separate test set, we constructed our own.

Our procedure consisted of the following steps:

1. We selected three sentences from the EMO-DB corpus and translated them into Italian (our native language) to ensure emotions could be conveyed as naturally as possible.

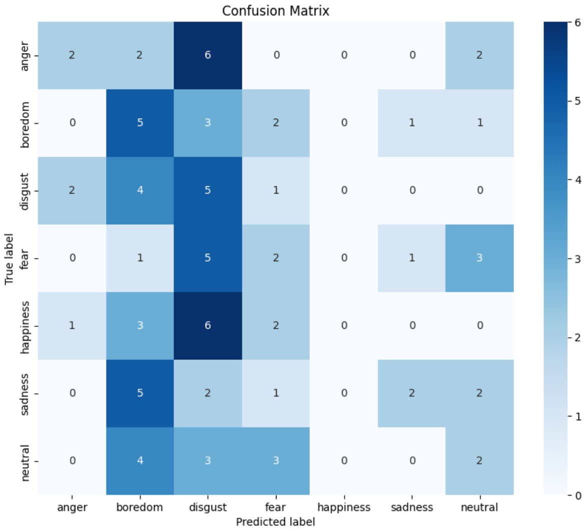
2. For each combination of sentence, emotion, and speaker, we recorded one audio file (with four speakers in total: the two of us and one acquaintance for each). Three speakers recorded the audio samples in Italian, while the fourth speaker (fluent in German) recorded them in German.

3. The audio files were saved in the same format as the EMO-DB data.

4. The recordings were preprocessed using the same pipeline as for EMO-DB, with the addition of standardization since EMO-DB files appeared to be standardized by default.

The result of this process is a preprocessed test set containing 84 audio recordings, with 12 samples per emotion. With this test set prepared, we implemented the final version of our model. We selected the best-performing architecture (CNN2D + LSTM), trained it on the entire EMO-DB dataset with data augmentation, and then evaluated it using our newly created recordings.

As expected, the results were less satisfactory than the performance observed during validation. On our test set, the model achieved an accuracy of 0.24 and a top 3 accuracy of about 0.56.



| | anger | boredom | disgust | fear | happiness | sadness | neutral |
|---|---|---|---|---|---|---|---|
| anger | 2 | 2 | 6 | 0 | 0 | 0 | 2 |
| boredom | 0 | 5 | 3 | 2 | 0 | 1 | 1 |
| disgust | 2 | 4 | 5 | 1 | 0 | 0 | 0 |
| fear | 0 | 1 | 5 | 2 | 0 | 1 | 3 |
| happiness | 1 | 3 | 6 | 2 | 0 | 0 | 0 |
| sadness | 0 | 5 | 2 | 1 | 0 | 2 | 2 |
| neutral | 0 | 4 | 3 | 3 | 0 | 0 | 2 |

*Figure 11 - Confusion Matrix Test Phase*

Analysis of the confusion matrix (Figure 11) revealed that the model tended to classify most recordings as belonging to only two of the seven emotions: boredom and disgust. Conversely, none of the recordings were classified as happiness.

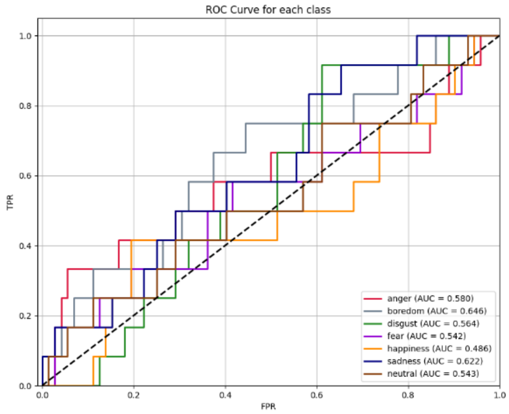


*Figure 12 - ROC curve Test Phase*

Examining the ROC curves and AUC scores (Figure 12), we observed that none of the emotions were classified particularly well, although all of them achieved an AUC above 0.5, except for happiness.

We anticipated lower performance compared to the LOSO cross-validation results; however, the drop was greater than expected. To better understand this outcome, we analysed the situation and identified several possible explanations:

- The language difference is a key factor: German and Italian differ significantly, and the way emotions are expressed in the two languages also varies. We attempted to address this issue by having one of our speakers record the sentences in German; however, this was likely insufficient, as the speaker was not a native German speaker.
- Another likely issue concerns the recording environment: while EMO-DB samples were collected in controlled settings using similar types of microphones, our audio files were recorded in ordinary home environments with different devices.
- Lastly, we must also consider our acting skills: none of the speakers in the test set are professional actors, and the emotions were staged according to our own interpretation, without prior experience or training.

Given these considerations, the test results appear less negative than they initially seemed. Nevertheless, we are not satisfied with the outcome; however, we chose to include the results for the sake of transparency, as they significantly affect the conclusions of our project.

## Conclusions

In this project, we aimed to build a deep learning model capable of classifying emotions from audio recordings, using the EMO-DB dataset as a starting point. The workflow followed a typical pipeline, including data preprocessing, exploratory analysis, feature extraction, data augmentation, model training, validation and selection, and finally, testing.

The preprocessing phase converted the raw .wav files by standardizing their length, frequency and sample rate. Feature extraction was then applied to transform the audio into Mel-Frequency Cepstral Coefficients (MFCCs), which represent an effective input format for deep learning models in audio processing.

Through data augmentation and Leave-One-Speaker-Out (LOSO) cross-validation, we enhanced models' performances and improved the robustness of our evaluation methodology.

During training, we experimented with several architectures, most of which achieved comparable results. For this reason, we focused our reporting on the MLP baseline and the two best-performing models: CNN2D and LSTM+CNN2D. Among these, the LSTM+CNN2D model clearly outperformed the others, and was therefore selected for the testing phase.

However, when evaluated on our newly recorded audio samples, this model performed significantly worse than in the validation phase, achieving much lower results.

Summing up, our final model performed extremely well in classifying the seven emotions within the EMO-DB dataset during training and validation, achieving an accuracy of 0.76, an average top 3 accuracy of 0.95, and AUC values ranging from 0.848 to 0.992 across the different emotions. However, it failed to generalize to recordings outside this context, particularly on the test audio files specifically created for this study and produced by non-professional speakers.

## References

Subramanian, R. R., Sireesha, Y., Reddy, Y. S. P. K., Bindamrutha, T., & Harika, M. (2021). Audio Emotion Recognition by Deep Neural Networks and Machine Learning Algorithms. IEEE Xplore. https://ieeexplore.ieee.org/document/9675492

ChatGPT. (n.d.). ChatGPT. https://chatgpt.com/

Doshi, K. (2021, February 12). Audio Deep Learning Made Simple – State-of-the-Art Techniques. https://ketanhdoshi.github.io/Audio-Intro/

Doshi, K. (2021, February 19). Audio Deep Learning Made Simple – Why Mel Spectrograms perform better. https://ketanhdoshi.github.io/Audio-Mel/

Doshi, K. (2021, February 24). Audio Deep Learning Made Simple – Data Preparation and Augmentation. https://ketanhdoshi.github.io/Audio-Augment/

Doshi, K. (2021, March 18). Audio Deep Learning Made Simple – Sound Classification, Step-by-Step. https://ketanhdoshi.github.io/Audio-Classification/

GeeksforGeeks. (2025, July 23). Mel-frequency Cepstral Coefficients (MFCC) for Speech Recognition. https://www.geeksforgeeks.org/nlp/mel-frequency-cepstral-coefficients-mfcc-for-speech-recognition/

GeeksforGeeks. (2025, July 23). Preprocessing the audio dataset. https://www.geeksforgeeks.org/data-analysis/preprocessing-the-audio-dataset/

Ma, E. (2019, June 1). Data Augmentation for Audio. Medium. https://medium.com/@makcedward/data-augmentation-for-audio-76912b01fdf6

Murel, J., & Kavlakoglu, E. (2023, November 16). What is regularization? IBM. https://www.ibm.com/it-it/think/topics/regularization