CNN Based Mel-Spectrogram Analysis and Speaker Recognition

January 20, 2026

Mattia de Luca (1883820), Tommaso Mattei (1884019)

Abstract

The following report presents the design and evaluation of a biometric speaker recognition system based on spectrogram representations and a deep learning model utilizing the ResNet architecture as a backbone. Speech recordings are processed by converting raw audio signals into mel-spectrograms, which preserve the distinctive spectral and temporal characteristics of individual speakers while providing a suitable two-dimensional representation for convolutional neural networks. These spectrograms serve as input to a ResNet model, enabling the extraction of discriminative voice features through deep residual learning and the use of the CosFace loss function. The proposed system is trained and evaluated on a voice dataset to assess its recognition performance. In accordance with standard biometric evaluation protocols, a probe test set is also used to examine the system's ability to identify speakers under realistic conditions. The results demonstrate the effectiveness and potential of the proposed approach for speaker recognition and highlight the applicability of deep convolutional architectures to real-world biometric voice recognition systems.

1. Introduction

In this work, we develop and analyze the performance of a custom ResNet-50 model which produces emebeddings for voice recognition. The experiments are conducted using audio samples from the VoxCeleb1 dataset, which consists of speech recordings collected from a large number of speakers and extracted from real-world video sources. Each audio clip is provided in .wav format and is labeled according to both the speaker identity and the original video segment from which it was obtained. Following common practices in the literature for this dataset, we adopt a spectrogram-based approach for audio representation. Specifically, speech signals are transformed into mel-spectrograms to capture discriminative time–frequency characteristics of individual speakers. Spectrogram-based representations have been shown to achieve comparable or superior performance in voice recognition tasks, as they make salient acoustic patterns explicit while remaining well suited for processing by convolutional neural networks. This representation allows the ResNet-50 architecture to effectively learn hierarchical features relevant to speaker discrimination.

Email:
Mattia de Luca <deluca.1883820@studenti.uniroma1.it>,
Tommaso Mattei <mattei.1884019@studenti.uniroma1.it>,
Github Repository < VoiceIdentificationSpectrogram>.

Biometric Systems 2025/2026, Sapienza University of Rome, 1st semester a.y. 2025/2026.

2. Related work

One of the main contributors for voice identification with spectrograms is the paper (Arsha Nagraniy, 2016) for the VoxCeleb1 dataset. Beyond their contribution with the dataset itself, they also were the ones to use a spectrogram based method with a custom made CNN. Another study (Prashant Anand1, 2019) explored the idea of using spectrograms with CNN, this time they also made a comparison between models, using ResNet50 as a base, showing promising results. One of the first studies using mel spectrograms instead of the base spectrograms is (Hao Chen, 2018), furthering the research on the subject. One final notable and very recent research paper is (Elliot Q C Garcia, 2025), which was followed as a guideline in terms of best practices to gather the most optimal results.

3. Method

Our research focused on the implementation of a voice recognition biometric system based on the ResNet architecture, modified to make it suitable for the task :

3.1. Dataset

VoxCeleb v1 is a large-scale audiovisual dataset introduced in (Arsha Nagraniy, 2016) for speaker recognition, collected automatically from YouTube videos of celebrities. It contains 1,251 speakers and over 100,000 utterances, recorded under unconstrained real-world conditions with significant variability in noise, recording quality, pose, and

illumination. Each utterance is associated with a corresponding face track, enabling audio–visual analysis. The dataset is widely used as a benchmark for speaker verification and identification due to its scale, diversity, and standardized evaluation protocols. In our work we chose to focus on the audio section of said dataset, where data was available in the format of a series of folder representing different users, all of which containing some internal folders which further subdivide the data into the original sources for the utterances, which are captured using the standard .wav format. Due to the large scale of the dataset and the limited hardware resources at our disposal we determined to only use a randomly selected subset of 763 People Of Interest, which allows for a much faster training at the cost of some accuracy. Due to the nature of such a choice the exact effect of this decision could not be determined but we believe this decision to be necessary given the high number of samples in the dataset and the available hardware resources.

Table 1. VoxCeleb dataset statistics (Max / Avg / Min). POI: Person of Interest

Statistic	**Max / Avg / Min**
Number of POIs	1,251
Number of male POIs	690
Videos per POI	36 / 18 / 8
Utterances per POI	250 / 123 / 45
Utterance length (s)	145.0 / 8.2 / 4.0

3.2. Preprocessing

As previously stated, the original files were given in a .wav format, which is a high-quality audio format that stores sound as raw, uncompressed data, preserving full audio fidelity but resulting in large file sizes. These audio files are then processed in a manner resembling the one described in the (Elliot Q C Garcia, 2025) paper. The audio files are first reduced in order to remove any potential silence at the beginning or ending of a clip; this is done in order to ensure all audio segment used in training accurately represent the desired speaking characteristics without the risk of pollution from badly selected audio sources.

3.2.1. Data Augmentation

After the removal of silent sections, the segments duration was uniformed to a 10 seconds duration in accordance with the findings of (Elliot Q C Garcia, 2025), where the authors tested the results of the training on audios of different durations and. This was done in the following manner: for audio clips of a length above 10 seconds, a simple subset of said audio file was used in the training, while in the case of segments below the standardized length two options were present in most of the literature, the most basic one being padding, which consists of simply filling the data structure representing the user's voice with zeros. We did not, however, use this technique and went with a looping methodology which, according to the authors of the paper(Elliot Q C Garcia, 2025) improves the quality of the training data and allows the model to perform better after training.

3.2.2. Mel-spectrograms

After the previously mentioned preprocessing of the audio files, these are turned into 2D arrays known as spectrograms, which are graphical representations of a signal's frequency which when applied to an audio file returns a representation of the distribution of energy across time and frequency, with the former forming the x-axis and the latter forming the y-axis. Mel spectograms, unlike regular spectrograms, are based on a logarithmic scale, which both better represents the human perception of noise and are more robust to noise and chages in recording conditions, allowing for better performances when used in neural networks, more specifically due to the heightened significance of learned features because the frequency scale matches perceptual importance.

3.2.3. Resizing

In the analysis of (Elliot Q C Garcia, 2025), three different spectrograms sizes are analyzed, which led the authors to determine that the best size for the task is a 432x288x3 matrix, so in order to achieve this results the spectrograms were first resized using the opencv library, which resulted in a 432x288 two dimensional matrix, which was then expanded via the numpy stack method to obtain a third dimension as required by the chosen model.

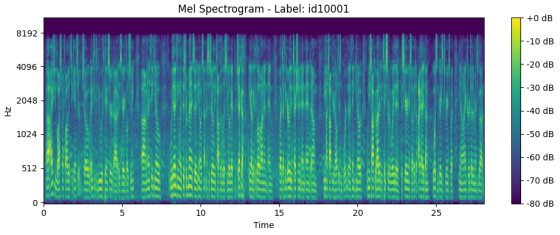

Figure 1. An example of a Mel-spectrogram obtained from the training set.

3.3. Model

The base of the model is Resnet50 (Kaiming He, 2015), a very popular CNN architecture, coupled with the CosFace loss (Hao Wang & Liu, 2018) instead of the standard softmax.

The last fully connected layer of the original model is substituted by a linear layer which reduces the dimensionality

of the embedding to 256 to which we then apply batch normalization. Instead of scores for the classes of a classification model, the final result is simply an embedding, or a series of embeddings with the batches, which represents the most relevant features of the input. Since this study is for speech/voice identification in an open set scenario, obtaining the embeddings is the key aspect necessary for the creation of the gallery and the comparison with the probe set.

3.3.1. Loss function

What truly differentiates our model from the baseline is the addition of the CosFace loss. As the name suggests this loss was created for identification through faces, but following the discoveries of (Elliot Q C Garcia, 2025) it's understood how beneficial it can be in the speaker recognition domain. For our task a simple softmax loss function would not be completely suitable, as seen in the first experiment in the literature over the VoxCeleb1 dataset. The reason for why this is the case is that softmax does not push the boundary between classes enough, resulting in a feature space in which different classes are still too close with each other. Beyond that we also want for the embeddings of the same subject to be close with each other, something that is not incentivised by softmax.

As it can be seen on Figure 2, the main idea for the CosFace loss is to have an angular margin between classes instead of being a singular point of contact like it would be for softmax. In order to do so, an hyperparameter called m is subtracted from the cosine similarity between features and target weights. This makes it so that classes are pushed apart and intra-class similarity rises. Within the actual calculation we also have an hyperparameter s which amplifies the separation between classes.

4. Metrics and results evaluation

In order to evaluate the system the test set was split between a gallery and probe sets. These sets contain data from novel users, who where hence not part of the original training set, and is split evenly between the gallery and probe set in order to achieve two sets of roughly equal number of audio segments for each user in the gallery, furthermore this was done in order to preserve the integrity of the original clips from which the audio segments were obtained, avoiding the splitting of data from a single clip between the two sets. This is done in order to avoid the model learning to extract information from contextual data that is not related to the speaker but rather to the specific environment where the audio was recorded. To achieve this a greedy load-balancing algorithm, where specifically a variant of greedy number partitioning (often described as the Longest Processing Time first heuristic, LPT). Each clip is assigned

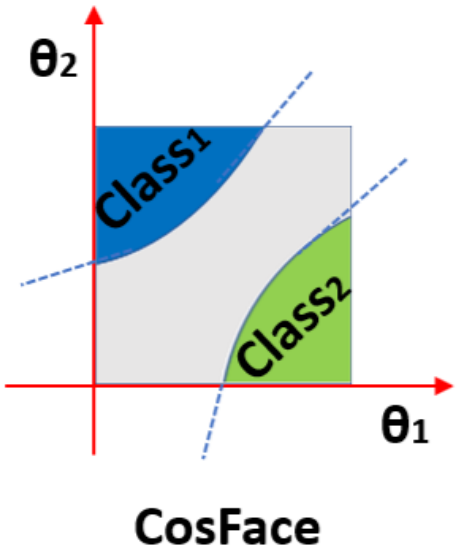

Figure 2. An example of how the cosFace loss behaves in the feature space

a weight k, equal to the number of segments (files) it contains and is then assigned to whichever set currently has the smaller total weight. This allows us to have an almost perfect balance between datasets in the context of our data with a time complexity of O(m*nlogn) due to the sorting required where n is the maximum number of segments in a clip and m being the number of clips.

Algorithm 1 Greedy Two-Way Load-Balanced Clip Splitting

Require: A set of clips $C = \{c_1, \ldots, c_n\}$, weight function $w(c)$
Ensure: Two subsets C_1 and C_2 with approximately balanced total weights
Initialize $C_1 \leftarrow \emptyset$, $C_2 \leftarrow \emptyset$
Initialize $W_1 \leftarrow 0$, $W_2 \leftarrow 0$
Sort clips C in descending order of weight $w(c)$
for each clip c in C **do**
  if $W_1 \leq W_2$ **then**
    $C_1 \leftarrow C_1 \cup \{c\}$
    $W_1 \leftarrow W_1 + w(c)$
  else
    $C_2 \leftarrow C_2 \cup \{c\}$
    $W_2 \leftarrow W_2 + w(c)$
  end if
end for
return C_1, C_2

Once the data was correctly divided, an inference was performed on each element of the gallery, the results of this operation were then averaged on a person to person basis.

This averaging allows us to perform a much quicker comparison between elements of the gallery and additional audio inputs at the cost of a lower accuracy when compared to methods such as the comparison between the segment we wish to perform a speaker recognition on and all elements from a user in a gallery, which could then be collapsed by averaging, or some other function. This method for recognition, however, suffers due to the much higher inference time and space cost, making it unfeasible for real time recognition on a large scale, but it does perform better accuracy wise in cases with a high intra class variation. This disadvantage of the chosen methodology can however be mitigated by normalization of the data within a gallery, which is what we decided to do in our recognition system. The elements of the probe set were then compared with all elements within the gallery using cosine similarity, which returned a float between -1 and 1 indicating the similarity between the features extracted from the spectrogram of the probe and the gallery elements, with 1 indicating a perfect match and -1 a maximally distant set of features. We then take into account the element of the gallery with the highest similarity to the probe element and apply a threshold to it. This threshold is used to determine whether the speaker is part of the gallery or not. In case this threshold is surpassed the identified user id is returned , if this isn't the case we determine that no match could be found. The experiments and subsequent results evaluation are obtained from a training with an embedding of dimensionality 256 and a training lasting 50 epochs with a learning rate of 0.001 on a NVIDIA 3070 Ti GPU.

4.1. Evaluation

The metrics used in our evaluation are accuracy, DIR (Detection and Identification Rate), FPIR (False Positive Identification Rate), FNIR (False Negative Identification Rate); uniting FPIR and FNIR we also have an open set ROC curve.

Before directly applying an arbitrary threshold over the final test with the gallery and the probe set, we created other sets made for validation, in order to avoid picking the best threshold directly on the test set and mirroring a real life scenario. Figures 3, 4 and 5 show the results of iterating through the validation set with a balanced number of known and unknown speakers.

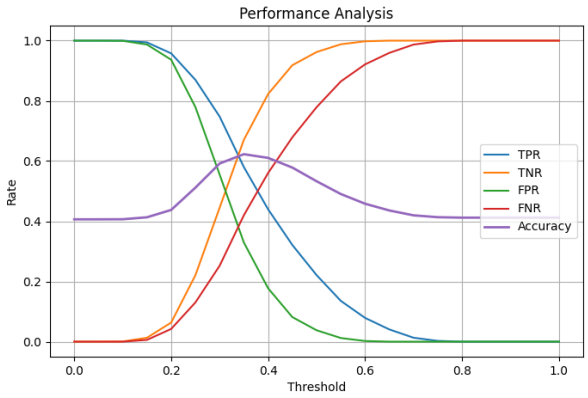

Figure 3. Accuracy and outcomes over different thresholds

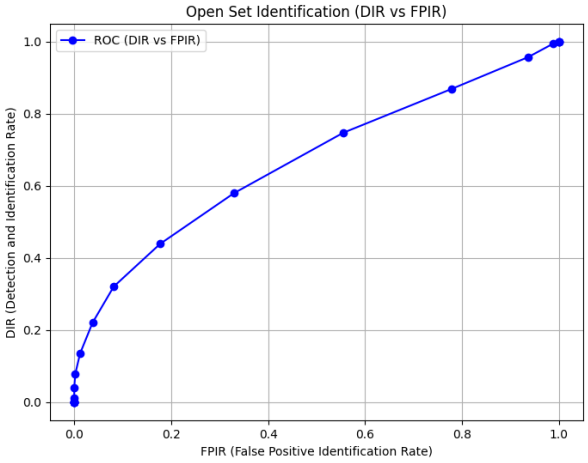

Figure 4. Open set ROC curve - DIR VS FPIR over different thresholds

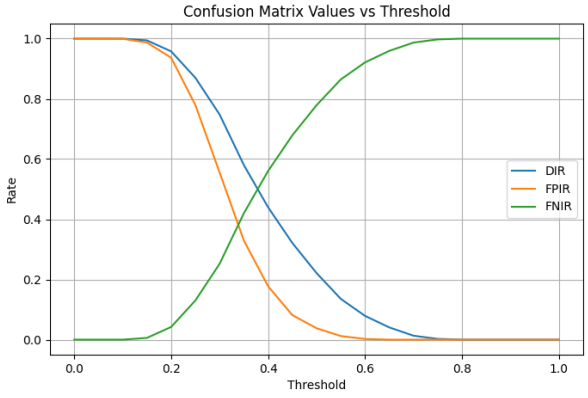

Figure 5. DIR, FPIR and FNIR over different thresholds

After looking at these results we settled on a threshold of 0.375, with a focus on lowering FNIR and upping DIR. With this in mind we tested on the actual gallery and probe set, receiving the following results:

Table 2. Performance Metrics

Metric	Value
Accuracy	0.7557
Detection Identification Rate (DIR)	0.5106
False Positive Identification Rate (FPIR)	0.1174
False Negative Identification Rate (FNIR)	0.4894

This kind of result can be considered moderate for several reasons. The accuracy is pretty good with a 75%, being in line with other results in the literature regarding speaker identification. If we were to simply look at the accuracy we could be satisfied, but by analyzing the DIR and FNIR we gather that the split between the rightful users being accepted and rejected is almost 50%. While this in a vacuum isn't particularly amazing, we must also remember the complexity of the audio domain with much more variance compared to other types of identification, and also the inherent complexity of an open set scenario. In comparison, the FPIR is around 11%, being very discriminative against unknown users, correctly rejecting most of them, showing how the results are apt and reasonable.

5. Discussion and conclusions

Our method showed how the advances in voice identifications are solid and that even when used with the ResNet50 model as a backbone they showed promising results. Through this study there were several hardware and time related complications which did not allow to explore the full scope of possible hyperparameters and optimizations. As a starting point, fully using the VoxCeleb1 dataset or even merging it with VoxCeleb2 could lead to important improvements in performance, especially taking DIR and FNIR in consideration. Even now the model shows the discriminative power of the CosFace loss function and how the preprocessed images managed to create significative embeddings. In the future, by refining and fine tuning even further the model, it should be possible to obtain better embeddings, capable of discriminating between classes even further.

Bibliography. (Arsha Nagraniy, 2016). (Elliot Q C Garcia, 2025). (Hao Chen, 2018). (Hao Wang & Liu, 2018). (Kaiming He, 2015). (Prashant Anand1, 2019).

References

Arsha Nagraniy, Joon Son Chungy, A. Z. Voxceleb: a large-scale speaker identification dataset. 2016.

Elliot Q C Garcia, Nicéias Silva Vilela, K. P. N. d. S. T. A. E. F. Text-independent speaker identification using audio looping with margin based loss functions. *arXiv:2509.22838*, 2025.

Hao Chen, Yusen Wu, P. N. C. L. Y. Y. Prosodic-enhanced siamese convolutional neural networks for cross-device text-independent speaker verification. *arXiv:1808.01026*, 2018.

Hao Wang, Yitong Wang, Z. Z. X. J. D. G. J. Z. Z. L. and Liu, W. Cosface: Large margin cosine loss for deep face recognition. *arXiv:1801.09414*, 2018.

Kaiming He, Xiangyu Zhang, S. R. J. S. Deep residual learning for image recognition. *arXiv:1512.03385*, 2015.

Prashant Anand1, Ajeet Kumar Singh1, S. S. B. L. Few shot speaker recognition using deep neural networks. 2019.