# Language Detection

**Data Mining, Machine Learning, and Deep Learning**

**Exam Project**

**MSc in Business Administration and Data Science**

| Student ID | Student name |
|---|---|
| 143003 | Diana Laura Janikowski |
| 158283 | Georgios Kotrotsios |
| 158798 | Andrea Pérez López |
| 158398 | Celine Schuhmann |

Submission date: 19. May 2023

Characters (incl. spaces): 34.240

# Abstract

Globalization and increased collaboration in international business have led to a greater demand for effective communication between individuals who speak different languages. Consequently, the significance of automatic language identification cannot be underestimated, as it plays a crucial role in facilitating communication and fostering understanding in multilingual environments. Researchers have been focusing on spoken language identification since the 1970s, and language identification systems have evolved over time. Traditional methods relied on identity vector systems, while recent approaches favor neural networks in combination with feature extraction techniques such as MFCC. In this project we build a spoken language identification system based on DNN and MFCC and examine its performance in distinguishing languages belonging to distinct language families. Specifically, we investigate the classification capabilities of our model in identifying languages from the Romance languages, German and Japanese, as no prior research has specifically addressed this comparative aspect.

**Keywords**

Supervised machine learning, MFCC, audio classification, language recognition, spoken language identification

# Table of contents

**Abstract** **2**

**Table of contents** **3**

**1. Introduction** **4**

1.1 Research question 5

**2. Related work** **5**

**3. Conceptual framework** **6**

Speech signals 6

Feature extraction 6

**4. Methodology** **8**

Data description 8

Pre-processing 10

Model 11

Model Architecture 13

**5. Results** **14**

Model_1 15

Model_2 16

Model_3 16

Model_4 17

Model_5 17

Model_6 18

**6. Discussion** **18**

**7. Conclusion** **20**

**References** **21**

# 1. Introduction

Increasing globalization and international business collaborations caused a rise in the demand for effective communication between people speaking different languages (Shahriar et al., 2020). The facilitation of communication and exchange of ideas among individuals is a pivotal function of language, with language identification serving as the initial step in multilingual environments (Venkatesan et al., 2018). As a result the need for automatic language identification (LID) becomes more prominent and its importance cannot be understated in regard to strengthening communication (Zissman and Berkling, 2001). Over 500 million people use Google Translate, however the application does not automatically detect the spoken language, but instead it has to be preselected which can cause problems as not everyone is able to manually detect the language they hear (Google, 2023; Shahriar et al., 2020). It is an impossible task for one person to become proficient in all languages but artificial intelligence technology, and more specifically machine learning (ML), can be taken advantage of and utilized to automatically identify the language a person speaks (Shahriar et al., 2020). Automatic LID involves the recognition of the language based on digitized speech samples (Zissman and Berkling, 2001). This process can be used in two ways: to assist machine-based systems or to aid human listeners. In the former, language identification can be used to facilitate a voice-activated travel information retrieval system that operates in multiple languages (Zissman and Berkling, 2001). Instead of running several speech recognizers in parallel, which can be expensive, language identification can be performed first, and then the appropriate speech recognition models can be loaded. In the latter case, language identification can be used to route an incoming call to a human operator fluent in the corresponding language (Zissman and Berkling, 2001). Language identification can save time and reduce the delay in finding a suitable interpreter, which could be crucial in emergency situations.

One of the approaches to building a LID system is teaching an ML algorithm what languages to take as inputs. When it comes to the type of algorithms that can be used for this task, Shahriar et al. (2020) mentions decision tree, random forest, linear regression and gradient boosting, where decision tree and random forest proved to be most successful. However, there are also other approaches to building LID systems based on identity vectors and neural networks (Bartz et al., 2017). Especially the latter combined with Mel-Frequency Cepstral Coefficients (MFCC), which is a feature extracting method, leads to state-of-the-art results.

### 1.1 Research question

The aim of the research conducted in this project is to contribute to the field of spoken language identification, ultimately making communication more efficient in a globalized world. While there are various research projects focused on building and comparing multiclass language classifiers in general, we have not come across ML research focusing on the impact of similarities in spoken languages. We therefore want to answer the question: how do similarities in languages affect the performance of a classification model? From a human perspective it is definitely easier to distinguish languages that come from different language families like for example Chinese and Spanish than languages that have the same roots like Spanish and Portuguese. This fact together with the experience that voice assistants, like SIRI or Alexa seem to take longer to adapt between similar languages made us suspect that this is impacting language identification tasks. Therefore we decided to take a deep dive into that direction and assess whether or not the accuracy of our model changes depending on the languages chosen.
For this purpose we chose languages from the Romance families and some unrelated ones like Japanese and English and compared results on different subsets..

## 2. Related work

Since the 1970s spoken language identification has been on the mind of researchers and language identification systems have evolved over time with traditional methods relying on identity vector systems for processing spoken languages (Bartz et al., 2017; Shahriar et al., 2020). More recently neural networks have gained popularity as feature extractors for language identification tasks. Specifically, Long-Short Term Memory (LSTM) networks have been favored for their accuracy and simpler design compared to traditional approaches (Bartz et al., 2017). The most effective approaches in language identification systems involve neural networks applied to input features. The features of speech can be categorized into two levels, namely low level and high level (Albadr et al., 2021). The low level includes commonly used acoustic, phonetic, phonotactic, and prosodic information. The high level, on the other hand, is dependent on the sentence syntax and morphology to identify the language (Albadr et al., 2021). The field of language identification mainly utilizes various acoustic feature extraction techniques, such as Linear Prediction Coefficients (LPC), LPCC derived from LPC, MFCC, and PLP. These methods are frequently employed to analyze and identify the characteristics of speech signals (Albadr et al., 2021). Various project authors have used several forms of LSTMs, such as unidirectional STM and Bi-directional LSTM (BLSTM), to process audio (Bartz et al., 2017).

## 3. Conceptual framework

**Speech signals**

The audio files are mapped as speech signals which are recorded observations measured based on the progression of time (Beigi, 2011). They can be perceived as the correlation between time and the intensity of speech waves during each particular moment in time. In general, signals can be stationary or non-stationary. In the case of the former, the statistical parameters such as intensity and variance do not change over time (Beigi, 2011). Speech signals serve as a prime example of non-stationary signals.

**Feature extraction**

Our chosen method of extracting features from audio signals is known as Mel-Frequency Cepstral Coefficients, and it was first introduced by Bridle and Brown in 1947 and later developed by Mermelstein in 1976 (Antony and Gopikakumari, 2018). “Mel” is a shortened form of the word “melody” and serves as a measure of pitch (Beigi, 2011). MFCC is a popular feature extraction technique in voice signal processing and is utilized for various applications, including speaker and voice recognition as well as gender identification (Abdul and Al-Talabani, 2022). This technique is based on the way the human ear perceives and processes sound. More specifically, human hearing does not process the entire audio range but instead focuses on specific areas and pays attention only to certain frequency components, allowing sound at specific frequencies to pass through while disregarding others that are unimportant (Cheng and Kuo, 2022). The human ear responds to frequencies below 1 KHz in a linear fashion, while frequencies above 1 KHz are perceived logarithmically. MFCC uses this property to analyze audio signals (Antony and Gopikakumari, 2018). In order to compute the MFCC, triangular Mel filters are employed in a process comprising five primary steps. The figure below depicts the process:

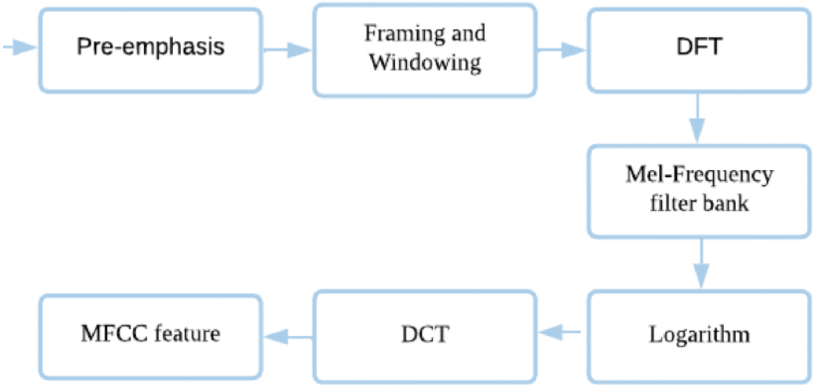


Figure 1. MFCC feature extraction (Abdul and Al-Talabani, 2022)

The first of the main steps is pre-emphasis which involves the enhancement of the signal quality which is achieved by the amplification of higher frequency signals (Antony and Gopikakumari, 2018). Those carry a higher weight in terms of discerning the signals as opposed to lower frequency. In the subsequent step, framing and windowing, the goal is to make the signal stationary to attain stable statistical properties which normally change with time as sound signals are non-stationary by default (Afrillia et al., 2017; Abdul and Al-Talabani, 2022). Windowing is an essential step in the calculation of MFCC features and it involves dividing the audio signal into short overlapping frames. The choice of the optimal window length depends on various factors, including the characteristics of the audio data and the specific task at hand. In our case we want to capture the similarities of spoken languages in short audio clips. According to Eringis and Tamulevičius (2014) windows between 7.5 to 10 ms in the highest speech recognition rate. In the third step the power spectrum is computed. Here, Discrete Fourier Transform (DFT) is often taken advantage of in order to take a look at power distribution among the frequency components that make up a signal (Abdul and Al-Talabani, 2022). In the following step, Mel frequency warping is performed with the use of filter-bank which is developed using pitch perception as a foundation. The filter bank aims to extract a nonlinear portrayal of the speech signal, similar to how the human auditory system interprets speech

(Abdul and Al-Talabani, 2022). The subsequent steps involve the application of Discrete Cosine Transform (DCT), originally suggested in 1972 by Nasir Ahmed, on the aforementioned filter bank which results in final MFCC feature extraction. According to Abdul and Al-Talabani (2022), DCT is used to "select most accelerative coefficients or to separate the relationship in the log spectral magnitudes from the filter-bank". Based on the described steps the speech signals are transformed into Mel spectrum and the end result of the feature extraction process is a Mel spectrogram (Cheng and Kuo, 2022):

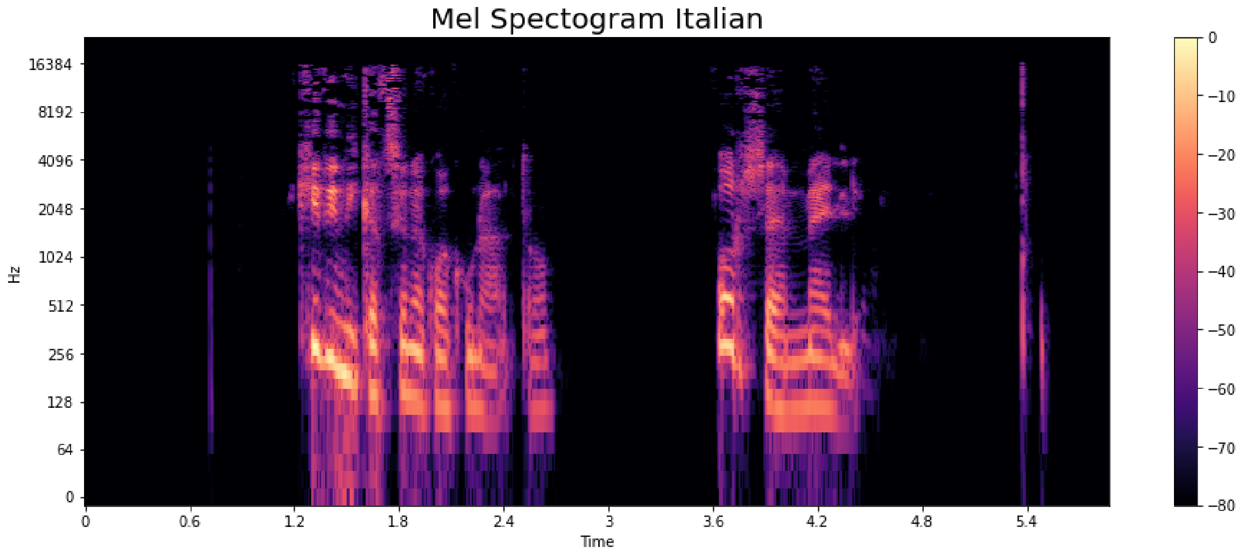


Figure 2. Spectrogram example (own product, 2023)

The Mel spectrogram depicts the power distribution of the audio, whereas the Mel spectrum is a numerical representation of the speech signal's features.

## 4. Methodology

### Data description

The dataset used, which is accessible here, comes from Common Voice which provides public voice datasets that can be used for machine learning models (Common Voice, 2023). The audio is recorded by volunteers worldwide which is afterwards validated by other people. Therefore it varies in quality, dialects and pronunciation as can be expected in a real-live situation. The audios that are provided in an mp3 format are relatively short with the majority of 3.5 seconds to 10 seconds. While longer audios are beneficial for improving the accuracy in most real-life applications, being able to make the language identification quickly is useful. It allows the program to continue with subsequent tasks like translating or connecting a caller to the right operators. Those characteristics make the data a realistic training set when the goal is to quickly recognize the languages. Furthermore higher computing power and time will allow to improve results by using bigger data samples for at least all common languages.

For the purposes of answering our research question we focused on two subsets of the languages available on Common Voice and used the same amount of samples (15 000) for each language to guarantee that the performance was not affected by the amount of data provided. As the familiar set of languages we chose languages from the Romanic language family, namely Spanish, Portuguese, Italian, Catalan and Romanian working with different subsets. For the control set consisting of languages that are perceived as easily distinguishable by humans the decision was made to go with German, Spanish and Japanese. The total dataset used in this project, therefore, contains 4.7 GB of audio speech files.

| Languages | Number of Clips | Largest Clip | Shortest Clip | Hours of Audio | Data size |
|---|---|---|---|---|---|
| Spanish | 17619 | 11.80 sec | 1.55 sec | 24.0 hours | 531 MB |
| Portuguese | 15033 | 10.58 sec | 0.072 sec | 16.4 hours | 373 MB |
| Italian | 17000 | 25.89 sec | 0.69 sec | 26.3 hours | 804 MB |
| Catalan | 17000 | 20.09 sec | 1.29 sec | 24.2 hours | 539 MB |
| Romanian | 17000 | 16.38 sec | 1.47 sec | 18.9 hours | 466 MB |
| French | 17000 | 11.06 sec | 0.91 sec | 22.8 hours | 855 MB |
| German | 15000 | 22.49 sec | 1.22 sec | 16.9 hours | 678 MB |
| Japanese | 17000 | 10.8 sec | 0.65 sec | 21.5 hours | 525 MB |

Table 1. Dataset details

**Pre-processing**

In the data preprocessing stage, our initial step revolves around handling '.tar' and '.tar.gz' files. The '.tar' extension, an acronym for 'tape archive', signifies a format that groups multiple files into a single larger file. The '.gz' extension, on the other hand, identifies files compressed with the 'gzip'. Consequently, the '.tar.gz' or '.tgz' file format is a fusion of 'tar' and 'gzip', resulting in a compressed tar archive. These files contain compressed collections of audio files. We use the built-in function that oversees the extraction process, with each '.tar.gz' file being decompressed into a new directory named after the original file. This step is crucial as it allows us to effectively access and organize the audio data.

Having the audio data readily available, the next phase involves exploratory analysis of the data. We employ the built-in function ,which is the core of the pre-processing process ,that samples the audio files and gives a detailed overview of the dataset content. This process uses libraries such as 'librosa', 'soundfile', 'matplotlib', and 'numpy' for operations like audio file reading, plotting waveforms, spectral representations, and computing the Mel-frequency cepstral coefficients.

In our case, we use the 'librosa' library to compute the MFCCs. 'librosa' is a Python library for music and audio analysis, providing the building blocks necessary to create music information retrieval systems. We use it to extract MFCCs for each audio file, which will be used as the input feature for our machine-learning model. It is highly important to mention that we set the window analysis to 10 ms, and the hop length is calculated by multiplying the window analysis and sample rate per clip.

The exploratory analysis also includes playing random audio files and plotting their raw waveforms and Mel Spectrograms. This is followed by computing the MFCCs and plotting the MFCC Spectrogram. The exploratory analysis concludes by providing a summary of each audio file including its sample rate, clip duration, total duration, and the duration of the largest and shortest clip.

In the next phase, we refine our dataset, taking into account the length of the audio files. By choosing files that don't exceed a particular duration, we're able to maintain a balance in the length of the audio clips. This strategy helps us avoid any potential bias in the model towards longer or shorter samples. Moreover, during the data pre-processing stage, we implement a rule to include a maximum of 15,000 clips for each language. This ensures equal representation of all languages in our dataset, mitigating any potential skewness or bias in our model.

For each audio clip, we repeat the audio to a fixed duration. This ensures that all audio clips fed into the machine-learning model are the same length.

Finally, for each audio clip, we extract MFCCs and add them to a Pandas DataFrame along with the corresponding language. MFCCs serve as the primary features of our model, while the languages act as labels. This data is then ready to be fed into a machine-learning model for training.

The data preprocessing stage serves as a bridge between raw data (mp3 files), and insightful analysis, through the extraction, exploration, and transformation of data.

### Model

The model finally used in the analysis is a Dense Neural Network (DNN) since it is an excellent choice for classifying spoken language based on MFCCs and resulted in the best outcomes. The characteristics of the Dense Neural Network makes it well-suited for this particular task. Its architecture is designed to effectively handle high-dimensional input data, such as MFCCs, which capture critical information about language phonetics. By leveraging its fully connected layers, the model can capture intricate relationships and patterns within the data. This architecture allows it to effectively learn and differentiate between the subtle linguistic nuances and patterns that distinguish different languages.

In this case, the Dense Neural Network is configured as a sequential model with multiple hidden layers and an output layer. This sequential architecture facilitates the sequential processing of input data, enabling the model to capture temporal dependencies and learn language-specific features. The number and sizes of the hidden layers is 3 and  was determined through experimentation to balance model complexity and optimize performance.

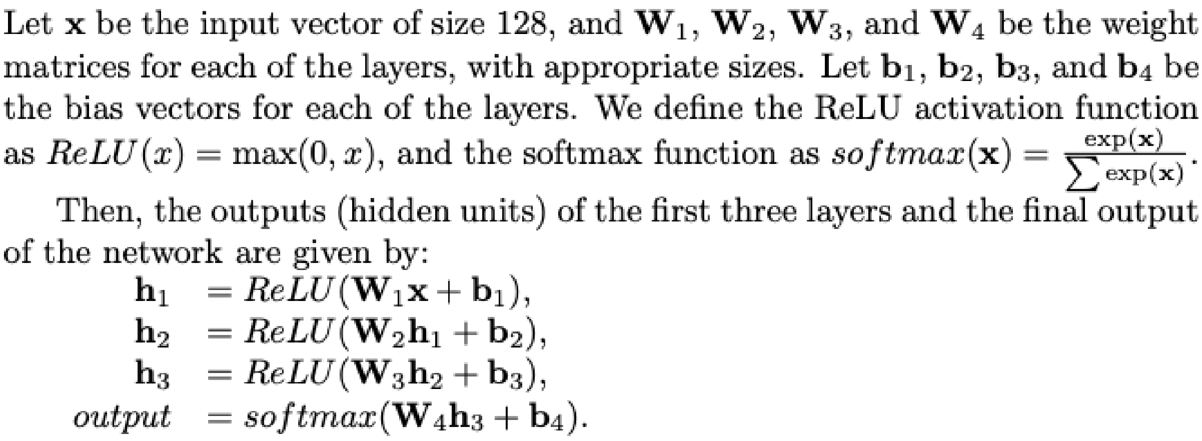
Activation functions play a crucial role in the model's architecture, enabling non-linear transformations and facilitating the capture of complex relationships within the spoken language data. By leveraging appropriate activation functions, the model can effectively discriminate between different languages based on their unique phonetic characteristics.

To enhance the model's generalization capabilities and mitigate overfitting,  a regularization technique called dropout was incorporated. Dropout randomly deactivates a fraction of the model's units during training, preventing the reliance on specific patterns or features and improving the model's ability to generalize to unseen data.

The final layer of the model is tailored to the multi-class spoken language classification task. The number of units in this layer corresponds to the number of languages being classified. Through the use of a softmax activation function, the model generates probability distributions for each language, providing a measure of confidence or certainty for each classification.

During the model training phase, an optimizer is applied to fine-tune the model's parameters and minimize the discrepancy between predicted and true language labels. The choice of optimizer is Adam, a  stochastic gradient method that is based  on adaptive estimation of first- and second-order moments.  This ensures efficient convergence and improves classification accuracy.

To quantify the model's performance during training and evaluation, a suitable loss function, such as categorical cross-entropy, is employed. This loss function measures the dissimilarity between the predicted language probabilities and the true language labels, guiding the model towards more accurate language identification.



Let $\mathbf{x}$ be the input vector of size 128, and $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{W}_3$, and $\mathbf{W}_4$ be the weight matrices for each of the layers, with appropriate sizes. Let $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$, and $\mathbf{b}_4$ be the bias vectors for each of the layers. We define the ReLU activation function as $ReLU(x) = \max(0, x)$, and the softmax function as $softmax(\mathbf{x}) = \frac{\exp(\mathbf{x})}{\sum \exp(\mathbf{x})}$.

Then, the outputs (hidden units) of the first three layers and the final output of the network are given by:

$$\begin{aligned} \mathbf{h}_1 &= ReLU(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1), \\ \mathbf{h}_2 &= ReLU(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2), \\ \mathbf{h}_3 &= ReLU(\mathbf{W}_3\mathbf{h}_2 + \mathbf{b}_3), \\ output &= softmax(\mathbf{W}_4\mathbf{h}_3 + \mathbf{b}_4). \end{aligned}$$

## Model Architecture

In our project, we have successfully implemented our model, which is accessible here, using the TensorFlow library with its Keras API. The Keras API in TensorFlow provides a simplified interface for creating and training neural network models, enabling rapid prototyping and experimentation with the following architecture:

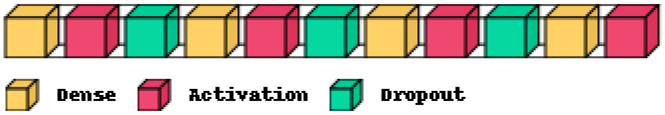


Figure 3. Image of the model created by visualkeras (own product, 2023)

Dense: This is the primary layer, which is fully connected. The first Dense layer has 100 units/neurons taking an input of 128 features (MFCCs), given the 12,900 parameters . The next Dense layers have 200 and 100 units respectively.

Activation: These layers apply the ReLU activation function to the output of the previous layer.

Dropout: This layer randomly sets a fraction of the input units to 0.5 at each update during training time, which helps prevent overfitting.

The final Dense layer has a different number of units, designed for a classification task from 2 to 6 languages. The final activation function is softmax, suitable for multi-class classification problems.

The total number of trainable parameters (53,503-53,806) is the sum of the parameters of all layers, which are subject to training through backpropagation. There are no non-trainable parameters in this network, meaning all parameters are updated during the training process.

## 5. Results

The model's performance is assessed using the accuracy metric on a separate test set. As a common evaluation metric for classification problems, accuracy measures the fraction of correct predictions made by the model. This is accomplished by applying the 'evaluate' method of the trained Keras model, which computes accuracy of the model on the test data.

Accuracy is a relevant metric to determine whether this model is suitable for our project. It provides a straightforward interpretation of the model's performance - the proportion of cases where the model's prediction aligns with the actual target label.

| Models | Languages trained | Reason | Computing Time | Accuracy Score |
|---|---|---|---|---|
| Model_1 | Portuguese, Spanish | Most similar languages | ≈25 min | 98% |
| Model_2 | Portuguese, Spanish, Italian and French | Most Spoken Romances Languages | ≈50 min | 84% |
| Model_3 | Portuguese, Spanish, Italian , French, Romanian and Catalan | Complementary Romance Languages | ≈85 min | 73% |
| Model_4 | Romanian, German and Japanese | Three total different family languages - Looking for if Romanian low | ≈45min | 90% |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  | accuracy is related to similarity |  |  |
| Model_5 | Spanish, German and Japanese | Three total different family languages - Testing Spanish to compare performance with Model_4 | ≈45min | 94% |
| Model_6 | Portuguese, Spanish, Italian | Most Similar Languages between the Romance Family | ≈50min | 98 % |

Table 2. Model’s summary

**Model_1**

Our initial model focused on the classification of 15,000 audio clips per language, Spanish and Portuguese, chosen due to their significant linguistic similarities. These similarities stem from their common roots in the Ibero-Romance group of languages, which results in overlapping vocabulary and similar phonetic patterns.

Despite these resemblances, the model successfully distinguished between the two languages with a remarkable accuracy rate of 98%. This suggests that even though Spanish and Portuguese are closely related languages, the model was able to identify subtle differences in pronunciation, intonation, and usage, allowing for effective binary classification.

Our success in this task illustrates the model's capacity to parse intricate linguistic nuances. However, it's also important to recognize that the high accuracy might be partly due to the binary nature of the task.

**Model_2**

In the development of Model_2, we extended Model_1 by incorporating Italian and French language data. However, this modification resulted in a decrease in accuracy from 98% to 84%. This decline is attributable to the less optimal classification performance for Italian and French, with accuracy rates of 76% and 71%, respectively. Notably, there was a significant rate of misclassification between Italian and French audio clips.

This confusion may be due to the inherent linguistic similarities between French and Italian. Like Spanish and Portuguese, French and Italian are both Romance languages and share a considerable amount of vocabulary, grammatical structures, and phonetic patterns. This overlap can make the task of distinguishing between them quite challenging.

Additionally, the decrease in accuracy could also be attributed to the increase in class complexity. As the model is now tasked with differentiating between four languages instead of two, the possibility of misclassification naturally increases.

**Model_3**

In Model_3, we further extended our previous models by incorporating two additional languages, Catalan and Romanian, bringing the total to six Romance languages. This adjustment led to a decrease in overall model accuracy to 73%. French and Romanian, in particular, showed lower individual accuracy rates of 56% and 64%, respectively, contributing to the overall dip in the model's performance. Nonetheless, the model maintained a high accuracy rate for Spanish at 98%.

A notable challenge arose from the classification problems among French, Italian, and Romanian. While Romanian might not seem as closely related to French or Italian in terms of vocabulary, it shares significant phonetic similarities with French, which could contribute to the observed misclassifications. These phonetic resemblances, coupled with the already mentioned similarities between French and Italian, added to the complexity of the task.

Increasing the number of classes, or in this case, languages, in a classification problem naturally raises the difficulty level and the potential for errors. As such, it's not surprising that as we expanded our model to handle six languages, its overall accuracy decreased.

### Model_4

In an effort to further investigate one of the lower-performing languages from Model_3, we conducted an additional experiment with Romanian, this time comparing it with languages from entirely different families, namely German and Japanese. This change led to a notable improvement in the model's accuracy score, elevating it to 90%. Despite this overall increase, Romanian's individual accuracy remained lower at 85%, compared to German at 91% and Japanese at 94%. This improvement in accuracy could be attributed to the reduced complexity of the task as we scaled down the number of languages from four to three.

Interestingly, the discrepancy in accuracy scores between Romanian and the other two languages might be a result of the inherent linguistic differences among them. Romanian, as a Romance language, shares more commonalities with other European languages in terms of vocabulary, grammar, and phonetics. On the other hand, German, while also European, belongs to a different language family (Germanic) and has distinctive syntactic and phonetic characteristics. Japanese, a language from an entirely different geographical and cultural context, possesses a unique writing system and phonetic structure, setting it significantly apart from Romanian.

### Model_5

In a final effort to determine whether Romanian is inherently difficult to classify, we replaced Romanian with Spanish, another Romance language, in the group with Japanese and German. This modification led to an increase in the overall model accuracy to 94%, suggesting that the model was indeed having a harder time distinguishing Romanian.

This result underscores the possibility that the distinct features of Romanian may not have been adequately captured by the MFCCs, and consequently, were not accurately detected by our model. This analysis highlights the need for further investigation into feature extraction techniques that can better capture the unique phonetic and linguistic characteristics of more complex languages like Romanian.

**Model_6**

Finally, after analyzing Spanish and Portuguese in Model_1 with an accuracy rate of 98%, and observing a decrease in accuracy to 84% upon the addition of Italian and French in Model_2, we decided to run the model without French, one of the languages contributing to the lower accuracy. In this iteration, the model regained a high accuracy level of 98%. This outcome suggests that Italian is more similar to French and Romanian than it is to Spanish and Portuguese, leading to the misclassifications observed earlier.

Furthermore, this exercise provides a valuable comparison with Models 4 and 5, as it involves an analysis of three languages from the same family, versus the analysis of three completely different languages in Models 4 and 5. Interestingly, even with the similarity within the Romance language family in this model, we still observed higher accuracy rates. This suggests that even among languages of the same family, there can be significant enough differences to facilitate high-accuracy classification. However, it also reiterates that the specific pairings and groupings of languages significantly affect model performance.

## 6. Discussion

Contrary to our hypothesis, the results of our research indicate that languages from the same family tend to be classified with higher accuracy in language identification tasks. This finding highlights the importance of differentiating between shared phonetic characteristics, such as frequencies captured by MFCCs, and linguistic characteristics based on language family tree theory. These differences should be considered when designing language identification models and assessing their performance. It is worth noting that the distinction between spoken and text-based language identification is also significant in this context.

One possible explanation for the increased performance of the model on similar languages is the presence of similar phonetic patterns specific to each language family. These patterns enable the model to effectively capture and discriminate between the languages, as it does not need to account for significant phonetic variation. Additionally, the proximity between similar languages plays a role, as closely related languages often share vocabulary, grammar, and linguistic structures. This proximity facilitates easier recognition and differentiation by the model.

Apart from inherent reasons, the dataset and model themselves can also influence the results. A more balanced data distribution for different languages can enhance the model's performance. Regarding the model, complexity plays a role, as a deep neural network with a large number of parameters can capture fine-grained distinctions between similar languages. The model can learn complex decision boundaries to effectively separate these languages. However, when dealing with different languages, the model may need to generalize across a broader range of linguistic variations, which may require a more sophisticated architecture or larger training data.

Our research emphasizes the importance of selecting the same set of languages when evaluating different models to improve comparability. Consistent languages across experiments enable better comparison and assessment of various models, providing deeper insights into their strengths and weaknesses in terms of different modeling techniques and architectures.

Furthermore, our findings suggest that certain languages may present inherent challenges for classification due to wider frequency and dialect variations. In such cases, subdividing languages with distinct accents, such as European Portuguese and South American Portuguese, could potentially enhance the overall accuracy of language identification models. However, further research is needed to explore the impact of language variation and dialects on classification performance before drawing conclusive findings.

It is important to acknowledge that our research has limitations. We focused solely on the Romance language family, and a broader comparison encompassing multiple language families would provide a more comprehensive understanding of performance variations. Additionally, a deeper background in language studies would help uncover additional factors influencing language identification accuracy, which could be combined with relevant metadata analysis.

Moreover, we recognize that the common Voice dataset may introduce biases due to the characteristics of individuals who contribute voice samples. These biases may limit the generalizability of our findings to the broader population. Future studies should aim to address these biases by incorporating diverse and representative datasets.

Another limitation of our approach is the simplification of complexity in computing. We calculated mean features of Mel-frequency cepstral coefficients (MFCCs) per audio clip for training. While this approach proved sufficient for some languages, it may not capture nuanced differences in others. Exploring alternative feature extraction methods or considering more advanced modeling techniques could potentially improve the classification accuracy for languages with greater variability.

Ethical considerations should be taken into account when applying language identification systems. Inaccurate detection of speech from individuals with certain mother tongues or less common dialects can lead to discrimination and hinder their access to crucial infrastructure or services. Therefore, it is essential to address potential biases and ensure that language identification models are fair, inclusive, and free from discrimination.

## 7. Conclusion

In conclusion, this research study focused on the utilization of deep neural networks (DNNs) for spoken language identification. A series of experiments were conducted to assess the performance of the models in classifying languages belonging to diverse language families and varying degrees of linguistic similarity. While our expectations were met in observing higher accuracy for a smaller number of languages, our research indicates that the impact of adding a language is contingent upon the particular language being included. Interestingly, contrary to our initial hypothesis, it was found that the DNN performs more effectively when classifying languages from the same language family, specifically the Romance language family.

This significant finding underscores the limitations associated with exclusively relying on human understanding to differentiate between languages. The decisive factor in automatic language identification lies in the phonetic features captured, particularly those derived from Mel-frequency cepstral coefficients (MFCCs). Furthermore, our investigation identified Romanian as a language that proves challenging to classify. This emphasizes the necessity of considering the specific languages under analysis and their unique characteristics to optimize the development of language identification systems.

## References

Abdul, & Al-Talabani. (2022). Mel Frequency Cepstral Coefficient and its Applications: A Review. *IEEE Access*, *10*, 122136-122158. 10.1109/ACCESS.2022.3223444

Afrillia et al. (2017). Performance Measurement Of Mel Frequency Ceptral Coefficient (MFCC) Method In Learning System Of Al- Qur'an Based In Nagham Pattern Recognition. *Journal of Physics: Conference Series*, *930*(012036). 10.1088/1742-6596/930/1/012036

Albadr et al. (2021). Extreme Learning Machine for Automatic Language Identification Utilizing Emotion Speech Data. *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*. 10.1109/ICECCE52056.2021.9514107

Antony, & Gopikakumari. (2018). Speaker identification based on combination of MFCC and UMRT based features. *Procedia Computer Science*, *143*, 250-257. 10.1016/j.procs.2018.10.393

Bartz et al. (2017). Language Identification Using Deep Convolutional Recurrent Neural Networks. In D. Liu, S. Xie, Y. Li, D. Zhao, & E.-S. M. El-Alfy (Eds.), *Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14–18, 2017, Proceedings, Part VI* (pp. 880-889). Springer International Publishing.

Beigi. (2011). *Fundamentals of Speaker Recognition*. Springer.

Cheng, & Kuo. (2022). Machine Learning for Music Genre Classification Using Visual Mel Spectrum. *Mathematics*, *10*(23), 4427. 10.3390/math10234427

Common Voice. (2023). *Common Voice*. Common Voice. Retrieved May 18, 2023, from https://commonvoice.mozilla.org/en/about

Eringis, & Tamulevičius. (2014). Improving Speech Recognition Rate through Analysis Parameters. *Electrical, Control and Communication Engineering*, *5*, 61-66. 10.2478/ecce-2014-0009

Shahriar et al. (2020). Identification of Spoken Language using Machine Learning Approach. *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, 1-6. 10.1109/ICCIT51783.2020.9392744

Venkatesan et al. (2018). Automatic Language Identification using Machine learning Techniques. *2018 3rd International Conference on Communication and Electronics Systems (ICCES)*, 583-588. 10.1109/CESYS.2018.8724070

Venkatesan et al. (2018). Automatic Language Identification using Machine learning Techniques. *2018 3rd International Conference on Communication and Electronics Systems (ICCES)*. 10.1109/CESYS.2018.8724070

Zissman, & Berkling. (2001). Automatic language identification. *Speech Communication*, *35*(1-2), 115-124. 10.1016/S0167-6393(00)00099-6