Université d’Ottawa

Faculté de génie



University of Ottawa

Faculty of Engineering

**ELG5901 Electrical Engineering Master Project**

**Spoken Language Identification System**

**DEBI Program**

**Final Report**

# Student name Abdallah Mohamed Ragab ID 300327288

# Student name Kareem Khaled Waly ID 300327303

# Student name Mohamed Khaled Elesawy ID 300327237

# Student name Hosam Mahmoud Ibrahim ID 300327269

# Student name Eslam Abdelraheem Shabaan Khalaf ID 300327261

# Graduate Program Master of Engineering Electrical and Computer Engineering – Artificial Intelligence and Data Science Track (DEBI Program)

**Semester to Register Fall 2022**

**(Indicate: Year & Fall, Winter, or Spring)**

**Project Title Spoken Language Identification System**

**Version 01 fall 2022**

# Abstract

*The object of this thesis is studying the Spoken Language Identification System (SLIDS) is a machine learning toolkit for automated speech recognition. It offers a wide range of acoustic and language identification features, including speaker accent recognition, automatic translation, and voice print analysis.*

*Speech recognition systems are increasingly used for computational linguistics tasks such as Speech-to-Speech Translation or multilingual Automatic Speech Recognition (ASR) systems, and the SLIDS is considered the initial stage in Natural Language Processing (NLP) processes. Unfortunately, current approaches to these problems are either limited in the scope of applications or lack the necessary accuracy to achieve human performance.*

*In order to implement our proposed system to overcome these limitations , we developed some Machine Learning (ML) and Deep Learning (DL) algorithms for the analysis of raw audio data, in addition, we trained some Convolutional Neural Networks (CNN) Models after converted the audio data to Spectrum image to find the best model accuracy, after that deployed this in our mobile and web application.*

*Discussion of feasibility/criteria for design consideration: I would like to add that a key aspect of this proposed algorithm is the dataset for our case largest datasets containing audio data from multiple languages are needed to train and test the algorithms to ensure that they are sufficiently accurate to be used in practical applications.*

*The system was tested in parallel with development using some audio data that was randomly scraped from the YouTube to ensure that our model is generalized.*

**Table of Contents:**

Contents

[Abstract 2](#_Toc124681495)

[1. Introduction 7](#_Toc124681496)

[**1.1 Problem Definition** 7](#_Toc124681497)

[**1.2 Background** 9](#_Toc124681498)

[**1.3 Project Context** 17](#_Toc124681499)

[**1.4 Detailed Project Plan** 18](#_Toc124681500)

[2. Design Overview 19](#_Toc124681501)

[**2.1** **Requirements** 19](#_Toc124681502)

[**2.2** **Detailed Design** 20](#_Toc124681503)

[**2.2.1 prototype models.** 20](#_Toc124681504)

[**2.3**  **Implementation** 28](#_Toc124681505)

[**2.3.1 Convolution Neural Networks Models** 28](#_Toc124681506)

[**2.3.2 Machin Learning Models** 30](#_Toc124681507)

[**2.3.3** **Deep Learning Models** 31](#_Toc124681508)

[**2.3.4** **Web scrapping:** 33](#_Toc124681509)

[**2.4** **Testing** 33](#_Toc124681510)

[3. Overall Results and Analysis 36](#_Toc124681511)

[4. Deployment Plan 45](#_Toc124681512)

[5. Conclusions and Future Works 49](#_Toc124681513)

**Table of Figures:**

Figures

[Figure 1: The Proposed Input and the expected Output 8](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700607)

[Figure 2: Problem definition [1] 9](#_Toc124700608)

[Figure 3: CNN Architecture [2] 10](#_Toc124700609)

[Figure 4: Confusion matrix plot for each language identification [2] 11](#_Toc124700610)

[Figure 5: T-SNE Plot for the Resulting Language Distribution [3] 13](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700611)

[Figure 6:Multiclass Classifier Confusion Matrix [4] 14](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700612)

[Figure 7:Spectrogram Image Generated from an English Raw Audio Data [4] 14](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700613)

[Figure 8: LDA dimension score comparation on i-vector model. [5] 15](#_Toc124700614)

[Figure 9: LDA dimension score comparation on x-vector model. [5] 15](#_Toc124700615)

[Figure 10: Algorithm Diagram for SLIDS [7] 16](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700616)

[Figure 11: Multiclass Model Confusion Matrix [8] 17](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700617)

[Figure 12: ROC Curve [8] 18](#_Toc124700618)

[Figure 13: Proposed System Requirements. 21](#_Toc124700619)

[Figure 14: SLIDS Walkthrough Diagram. 22](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700620)

[Figure 15: VGG16 Architecture [9] 22](#_Toc124700621)

[Figure 16: Extra-Tree Model Architecture [10] 23](#_Toc124700622)

[Figure 17: MFCC DL Baseline Models 26](#_Toc124700623)

[Figure 18: Mel Spectrogram DL Baseline Models. 27](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700624)

[Figure 19: Custom feature vector DL Baseline Models 27](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700625)

[Figure 20: Comparison between the best three DL Baseline Models. 28](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700626)

[Figure 21: There is an Overfitting when the Data is Small. 28](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700627)

[Figure 22: Our Proposed System Process 29](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700628)

[Figure 23: Spectrogram Image Sample 30](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700629)

[Figure 24: Signal Audio Sample 31](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700630)

[Figure 25: Comparison based on ML Models 32](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700631)

[Figure 26: Conv1D Classification Report 33](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700632)

[Figure 27: Simple DL Models Output Results Comparison. 33](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700633)

[Figure 28: Web Scraping Example from Medium [12] 34](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700634)

[Figure 29: Accuracy of the Best Models on YouTube Data. 35](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700635)

[Figure 30: Number of Audio Files per Language 36](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700636)

[Figure 31: The total length for each language 36](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700637)

[Figure 32: The audio length distribution 37](#_Toc124700638)

[Figure 33: CNN Models Accuracy 38](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700639)

[Figure 34: Pretrained VGG16 Model Classification Report on 35,000 Audio Files 39](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700640)

[Figure 35: Bar plot Accuracy of CNN Models 39](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700641)

[Figure 36: Bar plot Accuracy of CNN Models 40](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700642)

[Figure 37: VGG16 Model Classification Report on the whole Dataset 40](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700643)

[Figure 38: Pretrained VGG16 Model Confusion Matrix on the whole Dataset 41](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700644)

[Figure 39: Pretrained VGG16 Model Classification Report on the whole Dataset 41](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700645)

[Figure 40: Pretrained VGG16 Model Classification Report on the whole Dataset 43](#_Toc124700646)

[Figure 41: Extra Trees Classifier Confusion Matrix on the whole Dataset 44](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700647)

[Figure 42: Extra Trees Classifier Classification Report on the whole Dataset 44](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700648)

[Figure 43: Conv1D accuracy vs validation curve 45](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700649)

[Figure 44: Conv1D Loss curve 45](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700650)

[Figure 45: Conv1D Confusion Matrix on the whole Dataset 46](#_Toc124700651)

[Figure 46: Conv1D Classification Report on the whole Dataset 46](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700652)

[Figure 47: Best Deep Learning Model Architecture. 47](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700653)

[Figure 48: Simple DL Model Confusion Matrix on the whole Dataset 48](#_Toc124700654)

[Figure 49: Simple DL Model Classification Report on the whole Dataset 48](#_Toc124700655)

[Figure 50: Conv1D Model Confusion Matrix on the YouTube Data 49](#_Toc124700656)

[Figure 51: Conv1D Model Classification Report on the YouTube Data 50](#_Toc124700657)

[Figure 52: Conv1D Model Bar Plot for Gender Misclassification on the YouTube Data 50](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700658)

[Figure 53: The Project Walkthrough 52](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700659)

[Figure 54: Proposed System Mobile Application GUI 53](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700660)

[Figure 55: Upload Audio Files 53](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700661)

[Figure 56: Predict Language 54](#_Toc124700662)

[Figure 57: Web Application GUI 54](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700663)

[Figure 58: Predict Language 55](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700664)

[Figure 59: Upload Audio Files to Web App 55](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700665)

**Table of Tables:**

Tables

[**Table 1: Comparison Between the SCNN & CNN-RNN Approach [2]** 11](#_Toc124700666)

[**Table 2: Results of the Embedding Model on the KALAKA-3 Dataset [3]** 12](#_Toc124700667)

[**Table 3:Results of Some Models with Difference Timeslot [3]** 12](file:///C:\Users\mm\Desktop\AIMsTechnology_GP_Fall2023.docx#_Toc124700668)

[**Table 4: Final Models Comparison. [5]** 16](#_Toc124700669)

[**Table 5: Project plan milestones** 19](#_Toc124700670)

[**Table 6: Deep Learning Model Architecture** 24](#_Toc124700671)

[**Table 7: ANN Model Architecture.** 25](#_Toc124700672)

[**Table 8: Comparison based on VGG16 Model** 30](#_Toc124700673)

[**Table 9: Accuracy for the combination feature engineering with the ML models** 42](#_Toc124700674)

**Table of Acronyms:**

Acronyms

|  |  |
| --- | --- |
| Abbreviation | Definition |
| **SLIDS** | *Spoken Language Identification system* |
| **ASR** | *Automatic Speech Recognition* |
| **NLP** | *Natural Language Processing* |
| **VA** | *Voice Assistance* |
| **API** | *Application Program Interface* |
| **SCNN** | *Stacked Convolution Neural Network* |
| **CNN** | *Convolutional Neural Networks* |
| **MVC** | *Model, View, and Controller* |
| **acc** | *Accuracy* |
| **GUI** | *Graphical user interface* |
| **ANN** | *Artificial Neural Network* |

# Introduction

*Nowadays, the widespread use of AI has established itself as a global trend that no developed economy and hardly any company can escape. AI will revolutionize the way people work, learn, communicate, consume and live. AI can be used to promote social inclusion and to enable the disabled, and people with limited language skills or limited mobility to participate in the world of work and social life as equally as possible. The rise of intelligent technology will profoundly change the structure of the world economy, there are nearly eight billion people worldwide are speaking 7,000 languages, and language is the crucial thing for people to communicate with each other, and with numerous data of different languages find a way into digital form, and with the increase in the amount of this data being published on the web, knowing the natural language of all these pieces of information is one of the most essential processes, and knowing these language is very difficult for any system to distinguish between all them through voice only. Also, SLIDS consider the first stage of NLP. Moreover, the main products of many major technology companies, including Google, Microsoft, Apple, and Amazon almost now include Voice Assistance (VA). The SLIDS is composed of three primary components: data collecting, feature engineering, and language classification. The availability of an appropriate database is a need for creating and analyzing a voice identification system. In the next few years, cognitive systems will significantly expand their range of services. which results in a critical needing for an accurate automatic system that understands, analyzes their words, and interacts with the clients as fast as possible, and that is what we aim to do in this project we will develop the most crucial stage in this process which was language identification system through voice.*

Text

Description automatically generated

Figure 1: The Proposed Input and the expected Output

## **1.1 Problem Definition**

*There is a long-standing problem with the SLIDS which was how can computers accurately identify the language of a given utterance. SLIDS is a crucial initial module of some practical applications to improve the user experience of ASR, VA, speaker diarization, or any voice-related AI models. There are several commercial and practical uses for Language Identification Systems. For instance, a system like this could be used as a preprocessor to recognize the language being spoken before translation starts. If a call can be routed to a fluent operator in a foreign language using an automatic language identification system, telephone companies will be better prepared to handle those calls. Rapid translation and language recognition can even save lives.*

*The proposed model will help the sponsor provide a more efficient service to their customer since their field of work is NLP, and they create solutions to better understand and act upon customer conversations and feedback that help companies increase their profits by better understanding their customers. They can use our proposed system as a stand-alone Application Program Interface (API) to accurately identify a spoken language from the given speech sample at any time length format or as an initial module to apply ASR or any voice-related AI models and we aim to identify the language via voice only, and we will be handling seven languages: Arabic, English, German, Spanish, French, Italian, and Portuguese. Our proposed system can be considered a stand-alone project or counted as an essential step for another systems. Our proposed solution takes several approaches, including generating spectrum images for use in language identification by fading them into a CNNs and generating signals audio data for use in language identification by fading it into ML and DL models. The learning outcomes we hope to learn new aspects like audio structure and audio pattern processing using ML and learning deeply about DL and its different advanced models and deployment a big project via different python platforms like (FastAPI, Flask), finally merge our project model with our deployment back-end then connect them with our GUI (Graphical user interface) front-end.*

A picture containing diagram

Description automatically generated

Figure 2: Problem definition [1]

## **1.2 Background**

1. **Evaluation Parameters [2]**

**Accuracy (acc) =**

**Precision =**

**Recall =**

1. **Literature Review**

Fig. 1. - 
            CNN Architecture
          *It's difficult to automatically identify the languages through audio files, the problem was how to give the system properly parsed voice utterances and grammar rules so that it can quickly and accurately identify the language [2], The authors aimed to classify between different 5 languages German, French, Spanish, Russian, English, and Chinese language using the EU speech dataset which was a collection of wave format that required a lot of preprocessing before the modeling phase, the domain they used for that was the spectrogram images which is a visually frequencies representation of the signal throughout time instead of audio files domain, and they implementing the Stacked Convolution Neural Network (SCNN) which was a stack of four CNN layers as shown in Figure 3, also used Convolution Neural Network with Recurrent Neural Network (CNN-RNN). Authors represented the audio files by spectrogram images of size 536\*678 with batch size equal 50 and Learning Rate (LR) equal 0.001 and they achieved an accuracy 80% using SCNN and the confusion matrix for this result represent in Figure 4, also they achieved accuracy 79% using CNN-RNN and the comparison between two models shown in Table 1 demonstrate that the authors proposed system had a high performance compared to the previous work on the same work.*

Figure 3: CNN Architecture [2]

Fig. 5. - 
            Confusion matrix plot for each language detection
          

Figure 4: Confusion matrix plot for each language identification [2]

**Table 1: Comparison Between the SCNN & CNN-RNN Approach [2]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Evaluation** | **Accuracy** | **Precision** | **Recall** | **Error** |
| **SCNN** | *80%* | *57.4%* | *97.13%* | *19.24%* |
| **CNN-RNN** | *76%* | *56.2%* | *96.12%* | *20.12%* |

*Language recognition applications that automatically classify an utterance based on the spoken language become of interest to many academic disciplines. In a variety of applications, Spoken Language Recognition (SLR) is used as a pre-processing step. VOXLINGUA107 [3] is a benchmark dataset in audio scrapping from YouTube video part, and used by doing some analysis on a document that has more than 3000 characters and it should be a more than 10,000 articles so that they can research that they would use it in making and searching for related audio results which created around 6600+ hours of audio from 107 languages for around 66 hours per language, also for their experiments they used KALAKA-3 dataset which consist of three main partitions, training data that consist of six different languages Galician, Catalan, Basque, Portuguese, Spanish and English, development and test data that consist of twenty one European languages from YouTube clips, and they applied two embedding experiments on this data one on the cleaned format and another on the noisy format, There are four evaluation sets: Empty-Closed (EC) and Empty Open (EO) handle the classification of the closed and open sets for the Greek, Italian, German, and French evaluation sets, respectively. Plenty-Open (PO) and Plenty-Closed (PC), which handle closed and open set classification for the six stated languages, and results of these experiments as shown in Table 2. Also, they used NIST LRE07 dataset which consist of fourteen languages, and they applied multiple models as shown in Table 3. As a conclude they applied multiple models with an x-vector feature extraction technique and the best one was CNN with led with acreage score equal four, and that was higher than the benchmark by 16% and the T-SNE plot of resulting language shown in Figure 5.*

Graphical user interface, application, table

Description automatically generated**Table 2: Results of the Embedding Model on the KALAKA-3 Dataset [3]**

*Table

Description automatically generated*

**Table 3:Results of Some Models with Difference Timeslot [3]**

*Chart, scatter chart

Description automatically generated*

Figure 5: T-SNE Plot for the Resulting Language Distribution [3]

*In this study [4], the Novetta company proposed a technique to detect the language from the audio by using a Residual Neural Network (ResNets), they put in a lot of effort in preprocessing raw audio data to generate a spectrogram from it to be as an input for the CNN model as shown in Figure 5. For this approach they used the Voxforge dataset, which was an open-source corpus containing clips in different languages with different genders languages that exist in this dataset are English, Spanish, French, German, Russian, and Italian. They used Fourier Transform to create spectrum images from the audio and then fed these images to Resnet with these images as input, they trained this model with several approaches. Firstly, as a binary classifier, and secondly as a multiclass classifier with these six different languages. Binary Classifier On 60,000 samples for Russian and English clips achieved 94% accuracy, and when the binary classifier increased to 100,000 samples achieved 97% accuracy, which means that the more data fed to the model the more accuracy you will get, and for the multiclass classifier across the six languages achieved 89% accuracy and the confusion matrix shown in Figure 6. The fact that all the data are derived from the same dataset may limit the extensibility of these conclusions, despite the fact that they are quite promising. As a result of the large range of properties that audio formats can have, like bit rate, sampling rate, and bits per sample, we might anticipate that clips from other datasets collected in other formats could cause the network to become confused.*

Graphical user interface, application, calendar

Description automatically generated

Figure 7: Multiclass Classifier Confusion Matrix [4]

Figure 6:Multiclass Classifier Confusion Matrix [4]

Background pattern

Description automatically generated

Figure 7:Spectrogram Image Generated from an English Raw Audio Data [4]

*The authors in this study [5] tried to build a spoken language identification system to help the Indonesian people who use regional languages communicate directly with foreign language users, allowing them to use their local languages while using certain technologies, the dataset that the authors used included the following languages Bahasa Sunda, Bahasa Jawa, and Bahasa Minang, and they have scrapped 200 minutes audio files for each language. Before the modeling phase the authors used their custom i-vector and* *x-vector feature extraction technique to compare the accuracy results of the two. and in the modeling phase the authors have built two classifiers which is PLDA and logistic regression, before the dataset got classified the authors have used feature reduction technique which is LDA to reduce the dimensionality of the data and to see how this will affect the results as shown in Figure 1 & Figure 2. The authors used equal error rate (EER) score to evaluate the performance of their models, from Table 4 it can be concluded that i-vector and x-vector has better performance when using logistic regression compared to PLDA classifier. the authors chosen the i-vector with logistic regression as the champion model with EER score of 6, 0.83 and 0 on different audio segmentations, 3s, 10s and 30s respectively.*

Chart, waterfall chart

Description automatically generated

Figure 8: LDA dimension score comparation on i-vector model. [5]

Chart, waterfall chart

Description automatically generated

Figure 9: LDA dimension score comparation on x-vector model. [5]

**Table 4: Final Models Comparison. [5]**

Table

Description automatically generated

*To serve the language identification system, the approach here [6] Phonexia company which is a leading company in voice analysis that solves everyday challenges through voice biometrics and speech recognition has the ability to handle more than 30 languages, and they suggested a model that would take the audio in a specific format and return a JSON file with the desired output as shown in Figure 10.*

*Diagram

Description automatically generated*

Figure 10: Algorithm Diagram for SLIDS [7]

*Unaffected by the speaker's gender, age, or even their own characteristics, spoken language identification (SLID) is the process of identifying the language that is recorded in an audio file [8], Using four different datasets, the authors attempted to classify 22 different English, Russian, Indonesian, Spanish, Arabic, Urdu, Mandarin, Turkish, Chinese, Portuguese, Japanese, Latin, Swedish, Pushto, Romanian, Korean, Dutch, German, Hindi, French, Portuguese, Tamil, Tamil, Thai, and Estonian languages. Kaggle (3 languages, audio data), Language Identification dataset (22 languages, text data), Common Voice Kaggle (16 languages, audio data), and Mozilla Common Voice (4 languages, audio and text data) and two main approaches the deep learning CNN approach on Spoken Language Identification dataset and the Bernoulli Nave Bayes approach on a language identification data set. The CNN model architecture is composed of five convolution blocks, each of which has a conv2d layer followed by batch normalization and finally a max pooling layer, and after the five blocks, there is a flattened, dense layer, batch normalization, dropout, and finally dense layer, and for the parameter: The number of epochs is 60. Batch size equals 32, the activation function ReLU, optimizer Adam, and the loss function was categorical cross-entropy where the activation output based on the Spoken language identification dataset, was Softmax, and the data were split into 73080 train samples and 540 test samples, with an accuracy of 98%. In the case of Word embedding Keras Loss for the Word Category-based cross-entropy Optimizer Adam's activation function, and the output layer Softmax Bernoulli Type Naive Bayes Kernel Naive Bayes function Bernoulli achieved 95% on the Language Identification dataset, but this approach was built entirely on text.*

Graphical user interface

Description automatically generated with medium confidence

Figure 11: Multiclass Model Confusion Matrix [8]

Chart, scatter chart

Description automatically generated

Figure 12: ROC Curve [8]

## **1.3 Project Context**

*According to the business requirements, we employed several technologies and platforms. We utilized Linux and Windows operating systems, as well as PyCharm, Visual Studio Code, Anaconda, Google COLAB, Kaggle, and AWS Cloud. Python and PyTorch were used to create our project. Furthermore, we used several python libraries such as FastAPi, sikit-learn, and Libros, the evaluation metrics Accuracy, Precision, Recall, F1-measures, software management was Docker container, and regarding the problem analysis and research we searched for suitable data for our problem used by other people in similar work also we scraped data from YouTube for testing phase, and for the design part we applied a scalability model that able to solve our problem on different source audio file sizes and publish the ease-of-use design that facilitate others to interact with our GUI. Our input will be audio record which sent to the docker container then the audio record will undergo the prepressing step (cleaning and feature engineering) after that the audio will enter the model then the model will send the prediction language through the docker to the user.*

*The project's roadmap includes five major steps: data collection and preparation, convert the audio files data to something that machines can understand and deal with it, include feature extraction and selection, model architecture building and evaluation metrics choosing, testing on beta model, finally deployment the champion model as a final release phase.*

## **1.4 Detailed Project Plan**

**Table 5: Project plan milestones**

|  |  |
| --- | --- |
| **Tasks** | **Hours** |
| 1. Discuss the project with the mentor. 2. Research approaches to relevant technologies 3. Plan project schedule and deliverables 4. Present Proposal | 30 |
| **Project Proposal Final Version – September 14** |  |
| 1. Collect the data. 2. Preprocessing the data (Cleaning and splitting the data) 3. Creating Spectrum images from the voice 4. Create basic model as baseline performance model (ANN Model) 5. Feature Engineer | 30 |
| 1. Create CNN model. 2. Evaluation and error analysis 3. Hyperparameter tuning. 4. Create basic API | 30 |
| 1. Demonstrate progress in terms of minimum viable prototype to project mentor. 2. Complete requirements analysis and complete evaluation criteria with the project mentor 3. Create presentation (Screenshots of current status, Requirements, Criteria) 4. Research for alternative methods in case of unsatisfactory results | 30 |
| **Minimum Viable Prototype Présentation – Octobres 30** |  |
| 1. Try other models. 2. Evaluation and error analysis 3. Hyperparameter tuning. 4. Comparing between different models | 30 |
| 1. Choose the champion model. 2. Create final API. 3. Deploy the champion model to cloud | 30 |
| 1. Demonstrate project in terms of complete but not completed release to project mentor. 2. Validate project results with the project mentor. 3. Finalize design. Finalize TODO list. 4. Create presentation (Screenshots of current status, Results, Design, TODO) | 36 |
| **Beta Release Presentation – December 18** |  |
| 1. Deploy the final version of the prediction model. 2. Final demonstration and delivery of results to project mentor. 3. Draft report to project mentor and projects coordinator. 4. Submit a final report. 5. Complete Project Mentor Evaluation Form, and Team Peer Review | 60 |
| **Final Report, Internship Evaluation, (Team Peer Review) – January 15** |  |
| **Total Hours** | 300 |

# 2. Design Overview

*In this section we will discuss all experiments, tools, and process we have done with a detailed step, and the architecture of the best model for each approach.*

## **2.1 Requirements**

*Spoken Language Identification System is considered one of the most challenging problems that face both the Artificial Intelligence and Speech Recognition communities at the current moment, both have the common terms of potential business value and societal impact and effect. Since the communication must be clear and responsive. While body language and the use of images can be used to bridge the language barrier, these ways of communicating lack those two factors. Written communication translation is a popular alternative form of communication that provides clarity but lacks speed. Verbal communication is both clear and fast, but translation is much more difficult because there is no reliable way to determine which language is being spoken. Further research into spoken language detection can assist in overcoming one of the most significant barriers to clear and rapid communication. Furthermore, The SLIDS is a computerized system that can identify the spoken language from voice only, our stakeholders aim to develop a model that takes an audio file and returns the language spoken langue in the audio, the model should be built in a docker container, and will be accessible throw API, also the model should have the ability to keep up with real-life performance. The primary objective of the Language Identification System is to accurately identify a spoken language from the given speech sample at any time length, and Figure 11 shows the whole system requirements, tools, and the process. Languages to be detected: Arabic, Italian, Spanish, Portuguese, German, French, and English.*

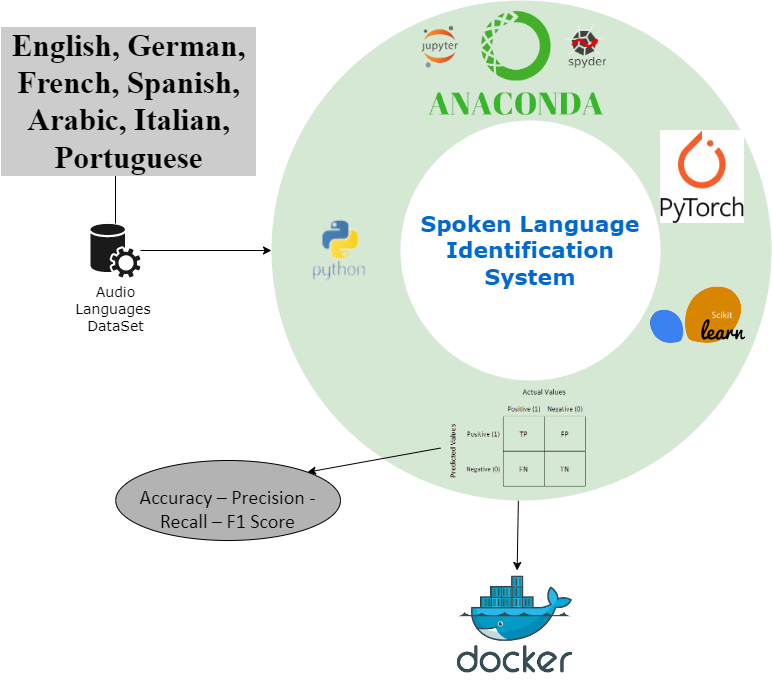
**

Figure 13: Proposed System Requirements.

## **2.2 Detailed Design**

*We divided system decomposition into three main approaches: CNNs, ML, and DL models with different feature engineering technique on different type of data, in this section, we will discuss the strengths and weaknesses of each of them.*

### **2.2.1 prototype models.**

*Our Study Process for the Spoken Language Identification System shown in Figure 14.*

*Graphical user interface, text, application, chat or text message

Description automatically generated*

Figure 14: SLIDS Walkthrough Diagram.

#### **2.2.1.1 Convolution Neural Networks Models**

*Classification images is a hard task because of the serval reasons as the dimension of the input will be huge for ML model or even normal DL model so there was an essential need to create a new approach to overcome these problems which called CNN which mimics the vision cortex of cats.*

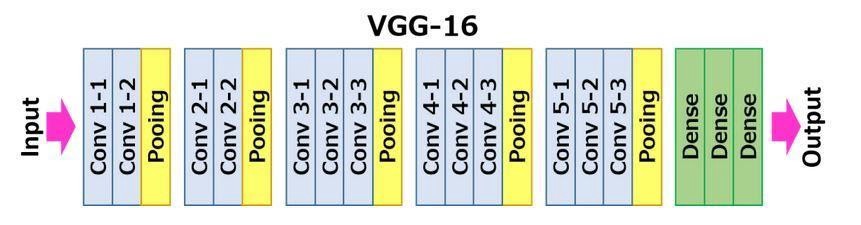


Figure 15: VGG16 Architecture [9]

*Starting with an earlier model like AlexNet which combination of convolution layers and max pooling and this model didn’t learn well as the problem was more complex compared to this model, moving to the VGG16 shown in Figure 12, which was built based on the AlexNet and replaced 5x5 filter with two 3x3 filters which decrease the number of parameters and making the learning process faster and it achieved the best accuracy, after that, we trying the ResNet which added residual blocks idea which added more layer and increasing the depth without causing overfitting by skipping some layers and by using the residual connection too, the test model was ResNet101 and it didn’t perform better than VGG16. Transfer learning is used in all models in this part which means transfer knowledge that the model learned on a specific problem on another problem in our case the models were trained on the ImageNet dataset.*

#### **2.2.1.2 Machine Learning Models**

*ML models have a tendency to overfit, which makes sense since the data had a large dimension space in the original state (the original signal and the MFCC and Mel-Spectrogram in the flattened state) or the model couldn't differentiate between the audios after the mean variance transformation. On the other hand, Random Forest Classifier and Extra Trees Classifier with the mean and variance of both the MFCC and Mel-Spectrogram could show a good result on the sampled data because both the meta estimator fits a number of decision trees on different sub-samples of the dataset. The Extra Trees Classifier shown in Figure 16, was the champion model when run on all of the data, but it did not provide satisfying accuracy.*

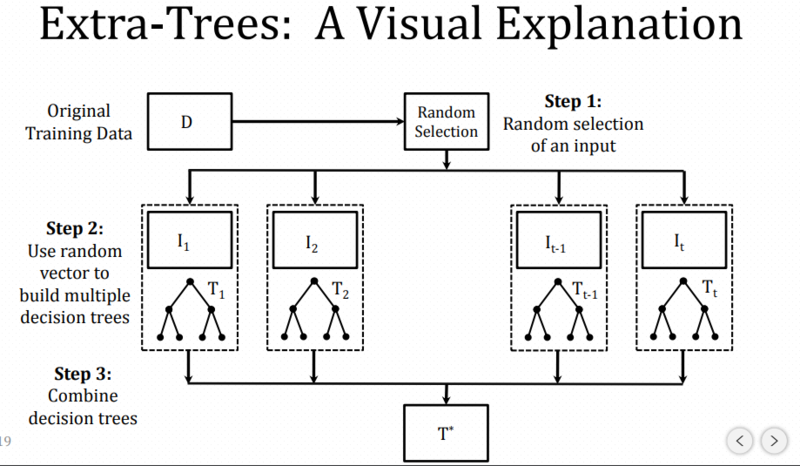


Figure 16: Extra-Tree Model Architecture [10]

#### **2.2.1.3 Deep Learning Models**

*Deep learning algorithms are capable of learning from their own mistakes, so they can be trained to adapt to new variations over time. We applied two different techniques based on the DL, which were as follows:*

* **Deep Learning with Image:**

*Since we are working with row audio data, then we can’t use the normal approach of having 2D convolution layers that are commonly used in dealing with images. cannot be used with audio, so 1d convolution (conv1D) was used instead. The model building can be split into two parts the stem and the stage. Also, the model had simple design principles: The Stem is consisted of a lazy conv1D followed by a ReLU activation function. As for the Stage part, it’s consisted of five repeated blocks. Each block is a combination of LazyConv1d and MaxPool1d this combination helps in maintaining the model's short and agile while containing the explosion of parameters. The model ends with a flattened layer followed by a dense layer, which acts as the final output layer. Since the data sample rate is 8,000 HZ and the fixed length of the audio data is 8 seconds, then the shape of the inputted data to the model is (BATCH SIZE equals 1, 64000), where 64000 is the number of samples rate multiple by the number of seconds, and the model architecture summary can be shown in Table 6.*

**Table 6: Deep Learning Model Architecture**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Main block name | Layer Name | # filters / kernel / stride | Output | # Of parameters |
| Stem | Conv1d  ReLU | (128,3,3) | [128, 26666]  [128, 26666] | 49,280  0 |
| Stage | (Convolutional Block 1)  Conv1d  MaxPool1d | (128,9,1)  (3,3) | [128, 26658]  [128, 8886] | 147,584  0 |
| (Convolutional Block 2)  Conv1d  MaxPool1d | (128,9,1)  (3,3) | [128, 8878]  [128, 2959] | 0  147,584 |
| (Convolutional Block 3)  Conv1d  MaxPool1d | (256,9,1)  (3,3) | [256, 2951]  [256, 983] | 0  295,168 |
| (Convolutional Block 4)  Conv1d  MaxPool1d | (256,9,1)  (3,3) | [256, 975]  [256, 325] | 0  590,080 |
| (Convolutional Block 5)  Conv1d  MaxPool1d | (512,9,1)  (53,3) | [512, 317]  [512, 89] | 0  1,180,160 |
| Flatten  Linear |  | [45568]  [7] | 0  318,983 |

* **Deep Learning with the signal data:**

*We have applied a simple DL sequential model based on a signal data to compared this with the DL model based on image data, the model architecture can be shown in Table 7, and to make sure which feature extraction technique will be suitable with the deep learning models in case of signal data, firstly we have built different baseline models for each feature extraction techniques which is: MFCC, Mel Spectrogram, and our custom feature vector which is combination of different features. Also, we have tried different samples sizes to see how this will affect the results. As it is shown in Figure 17 it is explaining the four different deep learning baseline models that we have built with the MFCC feature extraction technique. The same for the Figure 18 which explains the five different deep learning baseline models that we have built with the Mel Spectrogram feature extraction technique. As it is shown in Figure19 we have built three different deep learning baseline models using our custom feature vector which consist of different features, MFCC, STFT, chromagram, Mel spectrogram, spectral contrast and, tonal centroid.* *Figure 20 shows a comparison between the best three deep leaning baseline models for each feature extraction technique, our custom feature vector achieved the best results. These models trained on sample of data which is 35000 audio files (5000 audio files form each language). The two graphs in Figure 21 shows how different sizes sampling affect the training process.*

**Table 7: ANN Model Architecture.**

|  |  |
| --- | --- |
| Layers  With  Summary | Input layer (193) |
| Dense layer (8192) |
| BatchNormalization layer (8192) |
| Dropout layer (8192) |
| Dense layer (4096) |
| BatchNormalization layer (4096) |
| Dropout layer (4096) |
| Dense layer (2048) |
| BatchNormalization layer (2048) |
| Dropout layer (2048) |
| Dense layer (1024) |
| BatchNormalization layer (1024) |
| Dropout layer (1024) |
| Dense layer (512) |
| BatchNormalization layer (512) |
| Dropout layer (512) |
| Dense layer (256) |
| BatchNormalization layer (256) |
| Dropout layer (256) |
| Output layer (7) |

Diagram

Description automatically generated

Figure 17: MFCC DL Baseline Models

Diagram

Description automatically generatedDiagram

Description automatically generated

Figure 18: Mel Spectrogram DL Baseline Models.

Figure 19: Custom feature vector DL Baseline Models

Figure 19: Custom feature vector DL Baseline Models

Figure 18: Mel Spectrogram DL Baseline Models.

Chart, bar chart

Description automatically generated

Figure 20: Comparison between the best three DL Baseline Models.

Figure 20: Comparison between the best three DL Baseline Models.

Chart

Description automatically generated with low confidence

Figure 21: There is an Overfitting when the Data is Small.

Figure 21: There is an Overfitting when the Data is Small.

## **2.3 Implementation**

*Going through implementation, the programming language Python is used, and some important packages and libraries, such as OpenCV, PyTorch, and so on. The IDE used was local like PyCharm, VS Code or on the Cloud like google collaborator, and Kaggle. The Figure 22 represented all our work we have done.*

Diagram

Description automatically generated

Figure 22: Our Proposed System Process

Figure 22: Our Proposed System Process

#### **Audio Preprocessing**

*To preprocess the raw audio files and make them ready to be processed by both the ML, DL, and CNN models some basic preprocessing should be done. First, the audio files must all have the same length, so all audio files should be fixed to be 8 seconds long; if they are longer, the extra time will be disregarded. Also, the sample rate should be fixed for every audio file was less than 8 seconds long the audio will be padded with zeros and if it was longer the extra time will be discredited, also the sample rate should be Fixed for every audio to be 8,000 HZ, and all audio to have a mono channel.*

### **2.3.1 Convolution Neural Networks Models**

*Humans' voices transfer on the air, capturing the voice and representing it as a digital signal in format machines can understand and store in the memory, but applying DL models is still hard because of the large dimensions and the noise in the audio file. So, creating a new representation of the audio signal is needed and it was images called Mel spectrogram images representing the audio with an image as shown in Figure 23. Generating Mel Spectrogram images from the audio file was the first step and trying different CNN models using transfer learning was the second step. Models like AlexNet, VGG16, ResNet101, and MixVit, which is part of the vision transformer were tested, and the VGG16 had the best results due to increasing the complexity of the model and causing an overfitting problem. We already applied a fine tuned VGG16 and a pretrained VGG16 with Unfroze all layers and training it from scratch and in both cases it doing great compared to other CNN models as shown in Table 8.*

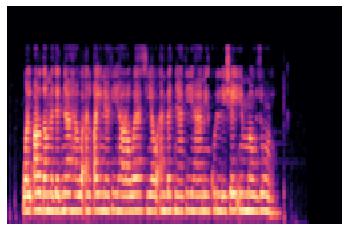
**

Figure 23: Spectrogram Image Sample

**Table 8: Comparison based on VGG16 Model**

|  |  |  |
| --- | --- | --- |
| Amount of data in minutes | Pre-Trained  VGG-16 | Fine-Tuned  VGG-16 |
| **50\_M** | 59% | 81% |
| **100\_M** | 62% | 83% |
| **200\_M** | 67% | 85% |

### **2.3.2 Machin Learning Models**

#### **2.3.2.1 Feature extraction**

Graphical user interface, chart

Description automatically generated*Applying ML to audio Data can be done using different ways like working with the audio data directly as a sequence of numerical values that represent the amplitude of the audio or working with it using some type of feature extraction for audio transformation like Mel-frequency cepstral coefficients (MFCC): commonly used features that are derived from speech signals for use in recognition tasks. seeks to create features from the audio signal that can be utilized for speech recognition that can identify phones and Mel Spectrogram to extract some useful data from the original audio data. But using these feature engineering with ML models would be hard since MFCC and Mel Spectrogram are 2D (multi-row for each data sample) features and classical ML models work with 1D (single-row for each data sample) data the data should be compressed somehow like for instance, taking the mean or variance of each feature data or flatten the data directly and feeding it to the model. Taking the mean and variance for the MFCC and Mel-Spectrogram will reduce the dimensions that we would be handling significantly.*

Figure 24: Signal Audio Sample

#### **2.3.2.2 Model selection**

*After creating multiple Feature Extraction techniques, we must combine them with different machine learning to search for the best combination to maximize the possibility of finding the best model, for the model selection five models were selected namely (HistGradientBoosting, DecisionTree, RandomForest, Bagging, ExtraTrees) classifiers to be combined with the Features selection namely (original signal, flatten MFCC, vertical mean of MFCC, vertical mean and variance of MFCC, flatten Mel-Spectrogram, vertical mean of Mel-Spectrogram, vertical mean and variance of the Mel-Spectrogram, vertical mean and variance of both MFCC and Mel-Spectrogram). As a result, when running on all dataset the champion model was the ExtraTrees classifier, but it didn’t give a satisfying accuracy as shown in Figure 25.*

*Chart, bar chart

Description automatically generated*

Figure 25: Comparison based on ML Models

Figure 25: Comparison based on ML Models

### **Deep Learning Models**

*Firstly, we are fixing the sample rate for every audio to be 8,000 HZ and fixing all audio to have a mono channel. As shown in Figure 26 the output from the conv1D model demonstrate that the model performance is good and with more training epochs can work perfectly. When it comes to models based on signal part, we first sampling the dataset to be 35,000 audio files equally distributed over all Languages, and the output shown in Figure 27.*

*Graphical user interface, chart

Description automatically generated*

Figure 26: Conv1D Classification Report

Figure 26: Conv1D Classification Report

*Chart, bar chart

Description automatically generated*

Figure 27: Simple DL Models Output Results Comparison.

Figure 27: Simple DL Models Output Results Comparison.

### **Web scrapping:**

*To test our best model on real world data, we have scrapped different audio files from YouTube using some tools like:* ***Selenium*** *used to create the Web driver, after that* ***BeautifulSoup*** *used to parse the Html page content to be able to read the YouTube videos links from the page.* ***pytube*** *used to download these YouTube videos links as an audio file.* ***AudioSegment*** *from* ***pydub*** *used to trim 10 random seconds from the downloaded audio files. For selecting the champion model, we tested the top 3 models on scrapped YouTube files and the results as shown in Figure 29, so We selected the Conve1d on the signal as the champion model because it generalizes better than other models and has the smallest size and inference time.*



Figure 28: Web Scraping Example from Medium [12]

*Chart, bar chart

Description automatically generated*

Figure 29: Accuracy of the Best Models on YouTube Data.

Figure 29: Accuracy of the Best Models on YouTube Data.

## **2.4 Testing**

**2.4.1 Data Plan**

*Initially, the data is Audio files collected from an open-source dataset called VoxForge which is an open-source speech dataset that was setup to be used in different Speech projects e.g., Speech Recognition, and Spoken Language Identification also it consists of more than 100,000 audio files with more than 240 hours. In addition to, Mozilla common voice, which is a dataset that contains 96 Languages, database contains a massive number of speakers approximately 271,807 Different speakers, and more than 20,817 hours of recorded audio clips of speech data also contains the voice of different male and female speakers with a variety of accents to eliminate any pronunciation bias from determining the language, and for the final test, YouTube videos that are legal to use (no copyright videos) were used to test the model on new distribution 700 audios were used distributed evenly across all languages. Aiming to train the model on different distributions the sample was taken from each dataset and combined together to generate the final dataset, the sample size was 25,000 audio files for each language from each dataset ending with 50,000 audio files for each language, and it was supposed to have a total number of audios equal 350,000 audio files but it’s only 333,575 in the final dataset as there were some language didn’t have perfectly 50,000 audio file which is shown in Figure 29, and Figure 30 shows the duration for each language in hours. Figure 31 shows the audio length distribution in seconds, and the selected audio length is 8 as the maximum length of the audio length.*

*Chart, bar chart

Description automatically generated*

Figure 30: Number of Audio Files per Language

Figure 30: Number of Audio Files per Language

Figure 29: Number of audio files per language

Chart, bar chart

Description automatically generated

Figure 31: The total length for each language

Chart

Description automatically generated

Figure 32: The audio length distribution

**2.4.2 Validation & Verification**

*For each step, testing output is important to transfer to the next one. All Model have been tested on the same data with the same length, the dataset was split into train, validation, and test with ratio 80% for train and 10% for validation and 10% for test and for final model selection the YouTube video were used, 700 videos which was scrapped as 100 video for each language and converted to audio then take a random 10 sec. from each audio we get a vary results on each model as shown in results sections. The accuracy metric is used to compare between the models.*

# 3. Overall Results and Analysis

*The major steps we follow are feature engineering, modeling, and prediction; we applied several approaches as we mentioned above:*

* ***CNNs approach:***

*When it gets to the CNNs model we applied the VGG16, Resnet101, MixVit, Deit-Tiny and Alexnet using the Voxforg datasets, and Mozilla common voice dataset, which included the seven language we aim to detected. Then we test best models on the dataset scraped from Youtube. The Dataset was split into train, validate, and test, and the evaluation metrics we all chose was accuracy. In contrast, we encountered some issues with training because of the computing power also a lot of the Portuguese audio data files are corrupted, so it was hard to find a 50,000 audio file to have a balance data.This part discusses the CNN models won Mel spectrogram images experiments result. First, a smaller sample from the dataset which is 35,000 audio files equally distributed across all languages after that Mel spectrogram images were generated from these sample, and different models have been trained and evaluated results are shown in Figure 33 the VGG16 and ResNet101 have the best accuracy in the transfer learning models and the Pretrained VGG16 has also a good performance as shown in Figure 34 and Figure 35. Then we trained the best three models which are VGG16, Pretrained VGG16 and ResNet101 on all datasets were the second step and results are shown in the Figure 33, from Figure we can demonstrate that VGG16 performed better rather than ResNet101 and final step were providing the classification report and the confusion matrix were shown in Figure 36, 37, 38 and Figure 39.*

Chart, bar chart

Description automatically generated

Figure 33: CNN Models Accuracy

Figure 33: CNN Models Accuracy

Graphical user interface, chart

Description automatically generated

Figure 34: Pretrained VGG16 Model Classification Report on 35,000 Audio Files

Figure 34: Pretrained VGG16 Model Classification Report on 35,000 Audio Files

Chart, bar chart

Description automatically generated

Figure 35: Bar plot Accuracy of CNN Models

Graphical user interface

Description automatically generated

Figure 36: Bar plot Accuracy of CNN Models

Figure 36: VGG16 Model Confusion Matrix on the whole Dataset

Graphical user interface

Description automatically generated

Figure 37: VGG16 Model Classification Report on the whole Dataset

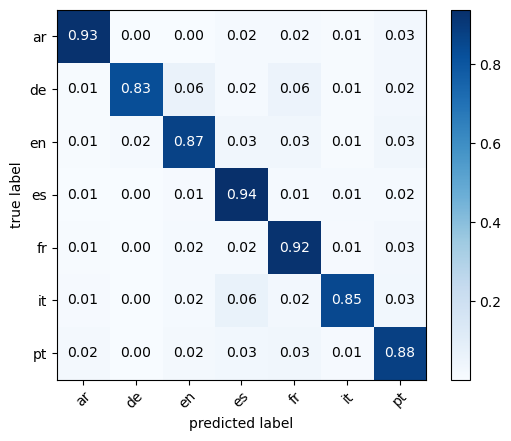


Figure 38: Pretrained VGG16 Model Confusion Matrix on the whole Dataset

Graphical user interface, chart

Description automatically generated

Figure 39: Pretrained VGG16 Model Classification Report on the whole Dataset

* ***Machine Learning approach:***

*In the ML approach, the work was divided into 2 parts feature engineering and modeling. For the feature engineering part: original signal, flatten MFCC, vertical mean of MFCC, vertical mean and variance of MFCC, flatten Mel-Spectrogram, vertical mean of Mel-Spectrogram, vertical mean and variance of the Mel-Spectrogram, vertical mean and variance of both MFCC and Mel-Spectrogram with the following ML models: HistGradientBoosting, DecisionTree, RandomForest, Bagging, and ExtraTrees classifiers.*

*As shown in Table 9 the accuracy for the combination of feature engineering with the ML models that ran on a sample of the data, which is around 25,000 audio files, we can conclude that for the Feature engineering vertical mean and variance of both MFCC and Mel-Spectrogram will be used because it gives more information, and the model can actually learn from it and to go to the point of running on all dataset we will choose the highest 2 models that produced the highest accuracy, Which is the Random Forest and Extra Trees. When running on the whole dataset the Extra Trees classifier continued to be the champion model with accuracy 60 % this low presenting can be the result of the huge size of the data since it’s around 330,000 audio files.*

**Table 9: Accuracy for the combination feature engineering with the ML models**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **HistGradientBoosting Classifier** | **DecisionTree Classifier** | **RandomForest Classifier** | **Bagging** | **ExtraTrees Classifier** |
| **Original signal** | 29 | 17 | 17 | 20 | 33 |
| **Flatten MFCC** | 45 | 30 | 44 | 38 | 47 |
| **Vertical mean of MFCC** | 41 | 30 | 41 | 37 | 43 |
| **Vertical mean and variance of MFCC** | 52 | 32 | 54 | 44 | 55 |
| **Flatten Mel-Spectrogram** | 14 | 15 | 14 | 14 | 30 |
| **Vertical mean of Mel-Spectrogram** | 51 | 39 | 55 | 47 | 57 |
| **Vertical mean and variance of the Mel-Spectrogram** | 63 | 44 | 62 | 44 | 69 |
| **Vertical mean and variance of both MFCC and Mel-Spectrogram** | 65 | 45 | 69 | 59 | 72 |

**Chart, bar chart

Description automatically generated**

Figure 40: Pretrained VGG16 Model Classification Report on the whole Dataset

**Chart

Description automatically generated**

Figure 41: Extra Trees Classifier Confusion Matrix on the whole Dataset

**Table

Description automatically generated**

Figure 42: Extra Trees Classifier Classification Report on the whole Dataset

* ***Deep Learning approach***
* ***Conv1D***

*In the DL approach we used the original audio signal and conv1d model to process the data, the model was trained on 100 epochs and with early stopping with patient 3 and it was applied on the validation loss, the accuracy and loss curve shown in Figure 43 and Figure 44, finally we provide the classification report and the confusion matrix were shown in Figure 45 and Figure 46.*

**Chart, line chart

Description automatically generated**

Figure 43: Conv1D accuracy vs validation curve

Figure 43: Conv1D accuracy vs validation curve

Chart, line chart

Description automatically generated

Figure 44: Conv1D Loss curve

Graphical user interface, application

Description automatically generated

Figure 45: Conv1D Confusion Matrix on the whole Dataset

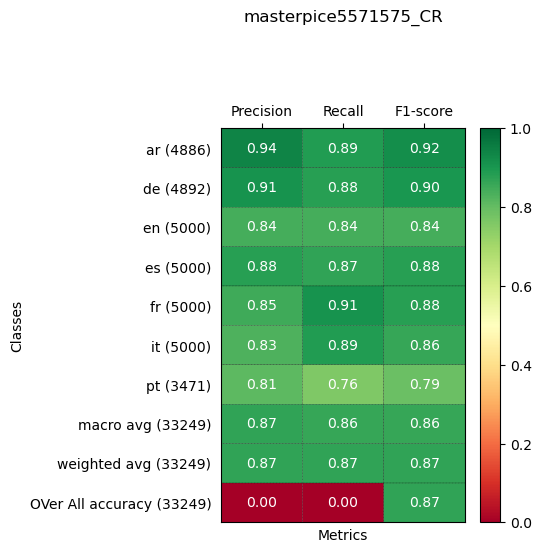


Figure 46: Conv1D Classification Report on the whole Dataset

* ***Simple Deep Learning Model***

*In the first phase, we’ve sampled the dataset to different sizes to see how this will affect the results and built different baseline models, after that on the second phase we’ve tried different ANN architectures, feature reduction techniques, and did hyperparameters tuning, and on the third phase after choosing the best ANN architecture shown in Figure 47 we’ve trained the model on the whole dataset which is 300,000 and it’s achieved testing accuracy of 84% as shown in Figure 48 and Figure 49, the training time was relatively slow because the architecture complexity, but adding more layers was likely increasing the accuracy but the training time will be too slow.*

Diagram

Description automatically generated

Figure 47: Best Deep Learning Model Architecture.

Figure 47: Best Deep Learning Model Architecture.

-> X6

**Calendar

Description automatically generated**

Figure 48: Simple DL Model Confusion Matrix on the whole Dataset

Chart

Description automatically generated with medium confidence

Figure 49: Simple DL Model Classification Report on the whole Dataset

* ***The best model results on the scrapped YouTube Data***

*we tested the top 3 models on scrapped YouTube data, which was VGG16, VGG16 from scratch, and Conv1D and the results shown in Figure 50, Figure 51, and Figure 52.*

Graphical user interface, application

Description automatically generated

Figure 50: Conv1D Model Confusion Matrix on the YouTube Data

A picture containing graphical user interface

Description automatically generated

Figure 51: Conv1D Model Classification Report on the YouTube Data

Chart, bar chart

Description automatically generated

Figure 52: Conv1D Model Bar Plot for Gender Misclassification on the YouTube Data

*This project helped us achieve learning outcomes for the graduate program throughout the whole engineering process, starting from the lecture review to learning more about the Spoken Language Identification problem, and what is the application based on this system and why it’s mandatory to any company to have one nowadays.*

*It was interesting to work with a unique field compared to classical NLP and CV which is audio analysis, it strengthens our knowledge in Computer Vision and Speech Recognition in parallel, and it was a good thing for us to know that represent the audio with an image and using transfer learning from domain to another domain can help to solve the problem, trying different approaches helps us in diving deep into the problem and gives us the possibility to discover interesting patterns in the data also testing the model on data from different distributions i.e. YouTube (gives us intuition /will indicate) how the model will face real-life problems, and we learned that In order to train a spoken language identification system, we need to extract relevant features from the audio data that can be used to distinguish between different languages, and also a spoken language identification system requires a large amount of audio data in order to be trained effectively. this data should be representative of the languages and dialects that the system will be identifying and should ideally include a wide range of speakers.*

*Working on this project gives us the experience of the real project as our sponsor is AIMs Solution. The objectives in the career perspective are:*

1. *Follow the road map and finish all assigned tasks before the deadline.*
2. *Build an agile methodology for project management.*
3. *Debug and solve errors to get the best possible results.*
4. *Teamwork and split the responsivities between us.*
5. *Prepare a weekly status report.*
6. *Keep in touch on a weekly basis with our uOttawa support and the company.*
7. *Get feedback and work on it.*

*The overall evaluation of the project's success and the outcome is excellent. We met all critical requirements which we discussed with our sponsor about.*

# 4. Deployment Plan

*To make our work useful for industry, we must deploy our work and make it publish and available for public ease-of-use. At first, we chose our champion model based on the accuracy metric from among all our work, then we used a python framework called flask, flask is a python framework used to develop restful API or develop web applications in MVC form (MVC refers to model, view, and controller, model refers to the database layer that communicates the database in our application, view here refers to the front-end layer that represents our pages and finally the controller playing as the backend layer all of these layers are communicating with each other in the same application, not in a separate application like restful API application). This flask application is used to load our champion model and make our preprocessing steps that must be done before giving our input data to the model, flask is based on python so our load model and many preprocessing steps that are written in python are not affected by such changes. In our application we used two forms of web applications first of them the MVC application which represents our web application as a stand-alone application and the other form is a restful application that works as a backend only to communicate with our mobile application. In the mobile application we used native android development using java programming languages and android studio IDE as a front-end mobile app and connected using the restful flask backend application.*

**

Figure 53: The Project Walkthrough

Figure 53: The Project Walkthrough

**The Mobile Application**

A close-up of a cell phone

Description automatically generated with medium confidence

Figure 54: Proposed System Mobile Application GUI

A close-up of a cell phone

Description automatically generated with medium confidence

Figure 55: Upload Audio Files

Graphical user interface, application

Description automatically generated

Figure 56: Predict Language

A picture containing diagram

Description automatically generated**The Web Application**

Figure 57: Web Application GUI

A picture containing diagram

Description automatically generatedGraphical user interface

Description automatically generated

Figure 58: Predict Language

Figure 59: Upload Audio Files to Web App

# 5. Conclusions and Future Works

*From the previous work, we can conclude that when identifying the language using the row signal data the normal ML approach couldn’t perform well because of the large dimensionality of the data and since it’s hard for ML to differentiate between the audio files, because the dataset has a lot of different accents and speakers, so it’s hard to identify the differences between the audios. On the other hand, DL models had better performance than ML models since the more data that were fed to the models the more the model had the ability to differentiate between the audio and find the pattern that the ML models couldn’t handle, also the huge diversity between the accents and speakers. On the other hand, when identifying the language using images training VGG16 from scratch provided better results rather than training with transfer learning as the transfer learning model has been trained on the ImageNet dataset which is not relevant to our project, furthermore, the model with transfer learning converged faster. Also, from the result of both the dataset test data and YouTube, it can be found that the champion model both in the image base or the raw signal-based approaches has a problem with the German language compared to the other language which is logical since the German language is lacking. Both the champion models in the two approaches -signal-based or image-based – have huge potential to provide a better result with further tuning for the hyperparameters or model Architecture design, and also solve the problem with the data.*

*In the future work we will search for more data in some language like Arabic and Portuguese to increase our training data, trying to solve the imbalance in the dataset, adding a threshold for the similar language from the beginning before start the training to the model can classify them accurately to overcome the misclassifying problem in the language that are have similar structure and punctuations rules. Also, to our model be more generalized we will be adding more dataset from different resources with different accents, and since each language was regarded as a separate module object, other languages from the same database can be added to the current model.*

References

[1] C. O'Sullivan, “Deep Neural Network language identification,” Medium, 21-Oct-2020. [Online]. Available: <https://towardsdatascience.com/deep-neural-network-language-identification-ae1c158f6a7d>. [Accessed: 13-Jan-2023].

[2] N. Bohra and V. Bhatnagar, “Language Identification using Stacked Convolutional Neural Network (SCNN),” 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, Jan. 28, 2021. doi: 10.1109/confluence51648.2021.9377037.

[3] J. Valk and T. Alumäe, “VoxLingua107: a Dataset for Spoken Language Recognition.” arXiv, 2020. doi: 10.48550/ARXIV.2011.12998.

[4] Deep Learning for Language Identification on Audio Signals (novetta.com)

[5] A. I. Abdurrahman and A. Zahra, “Spoken language identification using i-vectors, x-vectors, PLDA and logistic regression,” Bulletin of Electrical Engineering and Informatics, vol. 10, no. 4. Institute of Advanced Engineering and Science, pp. 2237–2244, Aug. 01, 2021. doi: 10.11591/eei.v10i4.2893.

[6] Voice Verification and Speech Recognition Software | PHONEXIA

[7] M. H, S. Gupta, D. Aroor Dinesh, and P. Rajan, “Noise-Robust Spoken Language Identification Using Language Relevance Factor Based Embedding,” 2021 IEEE Spoken Language Technology Workshop (SLT). IEEE, Jan. 19, 2021. doi: 10.1109/slt48900.2021.9383503.

[8] G. Singh, S. Sharma, V. Kumar, M. Kaur, M. Baz, and M. Masud, “Spoken Language Identification Using Deep Learning,” Computational Intelligence and Neuroscience, vol. 2021. Hindawi Limited, pp. 1–12, Sep. 20, 2021. doi: 10.1155/2021/5123671.

[9] R. G, “Everything you need to know about VGG16,” *Medium*. [Online]. Available: <https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918>. [Accessed: 14-Jan-2023].

[10] “Online Random Forest,” *Coding with food : 네이버 블로그*. [Online]. Available: <https://blog.naver.com/nywoo19/221621601124>. [Accessed: 14-Jan-2023].

[11] Sarthak, S. Shukla, and G. Mittal, “Spoken language identification using convnets,” *arXiv.org*, 09-Oct-2019. [Online]. Available: <https://arxiv.org/abs/1910.04269>. [Accessed: 15-Jan-2023].

[12] M. Singh, “How to scrape data from YouTube using python!,” *Medium*, 12-Apr-2021. [Online]. Available: <https://preettheman.medium.com/how-to-scrape-data-from-youtube-using-python-89f4f8ca4ddd>. [Accessed: 15-Jan-2023].

[13] “Mozilla Common Voice,” *Common Voice*. [Online]. Available: <https://commonvoice.mozilla.org/en/datasets>. [Accessed: 15-Jan-2023].

[14] “Welcome to flask¶,” *Welcome to Flask - Flask Documentation (2.2.x)*. [Online]. Available: <https://flask.palletsprojects.com/en/2.2.x/#api-reference>. [Accessed: 15-Jan-2023].

[15] “Overview,” *Docker Documentation*, 13-Jan-2023. [Online]. Available: <https://docs.docker.com/get-started/>. [Accessed: 15-Jan-2023].