THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

Data Science Program

Capstone Report - Spring 2024

Language Detection Systems Using Deep Learning Methods

Carrie Magee
Jack McMorrow

supervised by
Edwin Lo

Abstract

Language barriers pose a significant divide in today's diverse world, affecting everyday interactions between individuals and their communities. This capstone addresses linguistic divides through the creation of language detection and transcription systems. The primary objective was to leverage Transformer-based architecture models, specially Facebook's *Wav2Vec 2.0* and OpenAI's *Whisper* to develop a system that enables individuals to record their speech and receive real-time language identification and transcription. By using the Mozilla Common Voice dataset, our model was trained on 23 different languages, achieving an overall accuracy of 94 percent. The success of our project has the potential to promote inclusivity and accessibility in our multilingual communities.

Contents

1 Introduction

In our interconnected world, effective but accessible modes of communication are vital for making lasting connections, spreading information, and fostering understanding. Our everyday communities consist of individuals who speak numerous languages, exhibit various dialects, and possess different accents. Furthermore, as our world relies more and more on virtual communication it's important to address language barriers in virtual spaces.

If these communication barriers are not properly addressed they can have real implications in various domains, including in the classroom setting, where students from diverse linguistic backgrounds may face challenges comprehending course materials and thus, their ability to actively participate in the class. In our globalized business world, clear and seamless communication is essential for building strong professional relationships and achieving success. Similarly, in healthcare, providers often assist patients who speak different languages or lack the vocabulary required to advocate for their health needs. Language detection and transcription systems, which are the focal points of this capstone project, can improve patient-provider communication. This can lead to more accurate diagnosis and treatment, streamlining the patient experience, and potentially reducing interpretation costs.

Our project utilizes deep learning techniques to construct a fine-tuned audio classification and recognition model. More specifically, our models have the ability to identify and transcribe speech recordings in over twenty languages from only fifteen seconds of audio. The success achieved in this project contributes to effective and accessible communication, but also raises cultural awareness to various dialects and languages. In addition, the outcomes of the project can enhance user interactions in virtual settings and improve their experiences with online content,

ultimately contributing to a more immersive digital experience. By addressing language barriers through projects like this one, we are able to promote inclusivity, accessibility, and interconnectedness among individuals in our society.

2 Problem Statement

As previously mentioned, the problem addressed by our capstone project focuses on language accessibility. By creating a language identification and transcription system, our project attempts to bridge language gaps and increase accessibility in various facets of everyday life.

A significant challenge that plagued our initial progress involved the size of our dataset. The Common Voice 16.1 dataset by Mozilla, which was used to train our model, consists of over 3,000 hours of audio recorded in more than 100 languages. For the purpose of our project, we intended to train a multilingual identification model. Languages were included in the dataset if they featured over 250 hours of audio and maintained a validation rate of 75% or higher. This meant that at least 75% of the audio clips were assessed as accurate by native or proficient speakers of the respective language. However, the Common Voice dataset is formatted so each language is a subset of data as opposed to all languages in one dataset which further complicated our progress. Obtaining our desired dataset posed a significant challenge due to the computationally taxing nature of audio data and complex structure of the Common Voice dataset.

Another significant decision involved choosing the correct type of language processing task for our project. The type of task chosen for the project would ultimately influence our data preprocessing and overall model architecture. The two types of tasks best fit for our project were audio classification and automatic speech recognition. Audio classification involves assigning a label or class to a given sequence of audio inputs (“Audio Classification”, 2022). In contrast, automatic speech recognition (ASR) maps a sequence of audio inputs to a text output

("Automatic Speech Recognition", 2022). While both tasks involve analyzing audio sequences, automatic speech recognition focuses more on understanding the speech content within the audio, whereas audio classification focuses more on categorizing the audio itself. Considering our goal was to categorize audio based on the language being spoken, the latter option seemed to be the most appropriate choice for language identification while automatic speech recognition best fit for transcription.

Another challenge of the project concerned the type of model architecture to be used for each specific task. A common neural network architecture that is used for projects like these are called Transformers, which were originally documented in the paper *Attention is All You Need* (Vaswani et al., 2017). Transformers employ encoder-decoder architecture, where the encoder processes the input data and the decoder generates the output sequence (Vaswani et al., 2017). What makes Transformers unique is their implementation of self-attention mechanisms which capture long-range dependencies between input and output sequences, thus allowing the model to assess the input in its entire context, rather than one feature at a time. We will go into detail on the types of models that are common for audio projects like this.

3 Related Work

Deep learning has already been used extensively for audio classification and other audio related tasks, such as classifying audio signals for music, environment sounds, and speech, like we are doing here. Different types of deep learning networks can be employed on audio data, such as convolutional neural networks, recurrent neural networks, autoencoders, and Transformers (Zamin et al., 2023). Regardless of the method, there have been various ways to prove the effectiveness of deep learning for audio data.

Audio classification tasks began in the late 1990s when researchers proposed using Convolutional Neural Networks (CNNs) for audio classification, which have since been widely adopted for various applications, including speech recognition and audio classification. In these models, the one-dimensional interpretation of audio data is converted to a two-dimensional representation, and then inputted to the model (Zamin et al., 2023). These two-dimensional representations are usually spectrograms, or other two dimensional representations. Recurrent Neural Networks (RNNs) were first introduced by Graves in 2006. RNNs can be great for complicated data forms like audio, and help address certain issues with traditional neural networks, such as vanishing gradients (Graves et al., 2006). In RNNs, the output of the network is also inputted in the next iteration, allowing the model to recognize new features in context of previous elements that have already passed through. This allowed models to evaluate context, which can be critical for certain tasks with audio data. However, their memory tends to be short term, which led to the development of long short term memory (LSTM). Over the past decade, audio classification has been widely adopted in various types of models, including transformers, autoencoders, and hybrid models, which incorporate different combinations of other methods.

Autoencoders have also been adopted for audio tasks in deep learning. These models use an unsupervised learning technique. It starts with an encoder to transform the data into a lower dimensional format and then its followed by a decoder to reverse the output to the original format of the inputted data.These have frequently been used to classify audio data, due to the autoencoder's ability to handle their complex nature (Zamin et al., 2023). Transformers, on the other hand, have traditionally been used for text related data, but have since been adopted for uses in audio data. Transformers use self attention mechanisms that allow it to understand the input data in its global context. These models have been very promising for audio classification

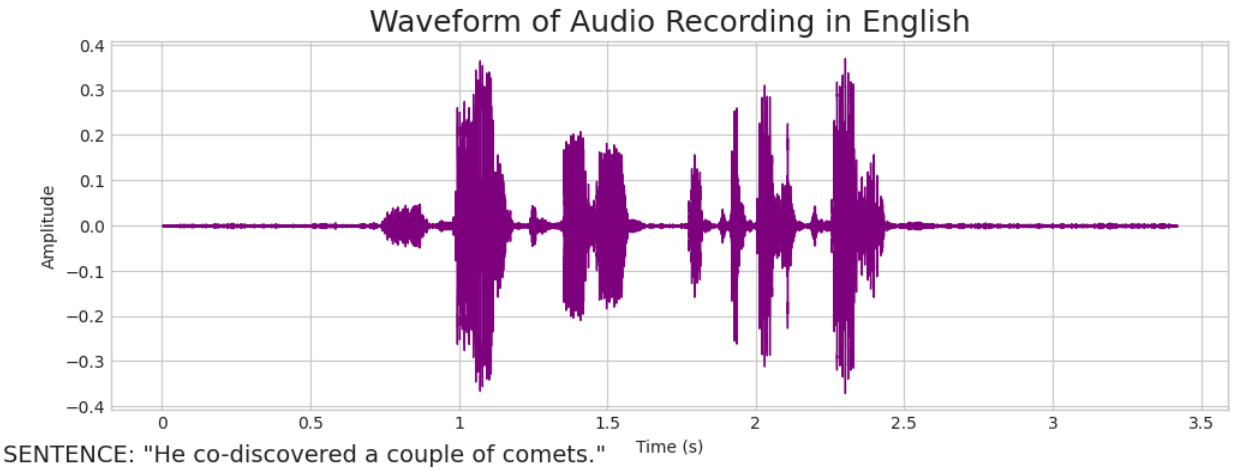
tasks, however they require incredibly large amounts of data in order for them to train properly, which results in limitations when computation ability is not an abundant resource for researchers looking to employ audio classification tasks (Zamin et al., 2023).

Overall, the past couple of decades have seen a dramatic boom in the research and applicability of audio related data. We plan to take advantage of this research for the development of our models for audio classification across many different languages.

4 Solution & Methodology

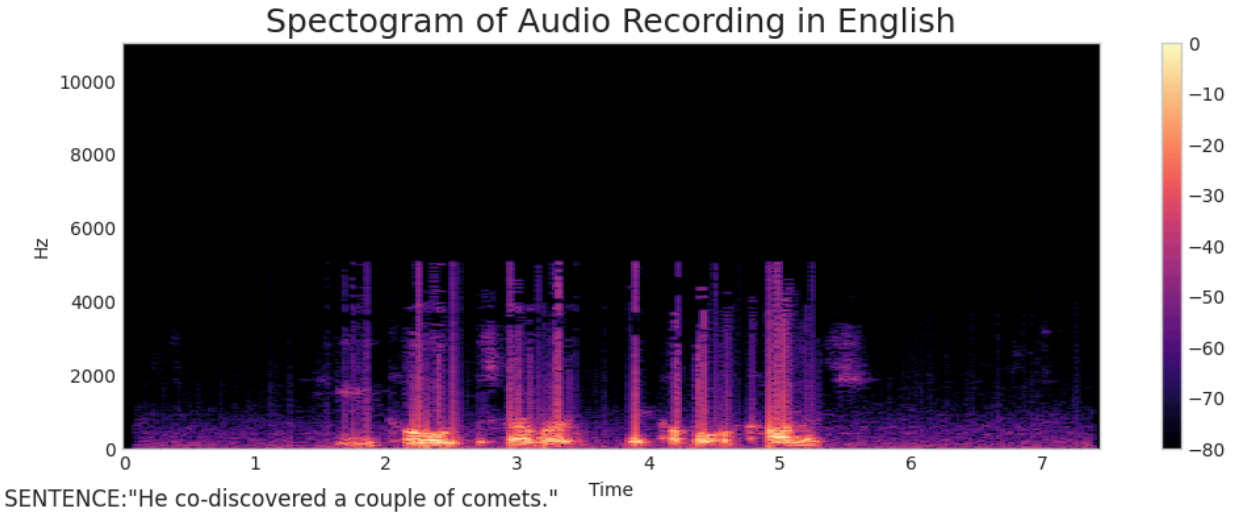
Audio data is unique, not only in its inherent structure, but also in the way it must be processed in order to be used in various deep learning tasks. Audio data consists of continuous sound waves which are a series of arrays that represent the air pressure at a certain moment in time. This sound pressure is known as amplitude (dB) and can influence the size of the audio data when converted into a usable format for training. These continuous sound waves need to be converted into a series of discrete values in order to be processed by digital devices. Sampling is the process of converting these continuous values into discrete waveform representations by gathering measurements of the continuous signal at fixed time intervals. The sampling rate is the number of samples taken per second (ie. Hertz) and is extremely important to consider when processing audio data for training models (Por et al., 2019). There are a few different ways to represent audio data. A common format is the waveform which shows changes in amplitude (loudness) over time as seen in Figure 1.

Figure 1. **Waveform Visualization of English Audio Recording**



After the audio has been digitized and resampled, feature extraction is the process of identifying and extracting relevant information from the raw audio signals, represented as waveforms, to serve as suitable inputs for models. The most common type of feature extraction is known as a spectrogram. Since a spectrogram displays changes in frequency of an audio signal over time, we can represent time, frequency, and amplitude at once. The spectrogram is crucial for the purpose of this project because languages vary in duration, frequency, phonemes, and acoustic properties which can be easily identified through spectrogram graphs (Figure 2). Spectrograms are crucial for understanding the properties of audio data, but they are not the only viable method.

Figure 2. **Spectrogram Visualization of English Audio Recording**



Mel-frequency cepstral coefficients (MFCCs), as illustrated in Figure 3, are a more advanced technique that builds upon spectrograms. They provide a detailed understanding of the frequency content of an audio signal over time, but on a logarithmic scale. The logarithmic scaling of audio input closely aligns with human auditory perception and makes it useful for tasks like speech recognition or audio classification. These distinctive properties of audio data influence the way in which we must process the data and can ultimately affect the outcome of our models if not transformed properly.

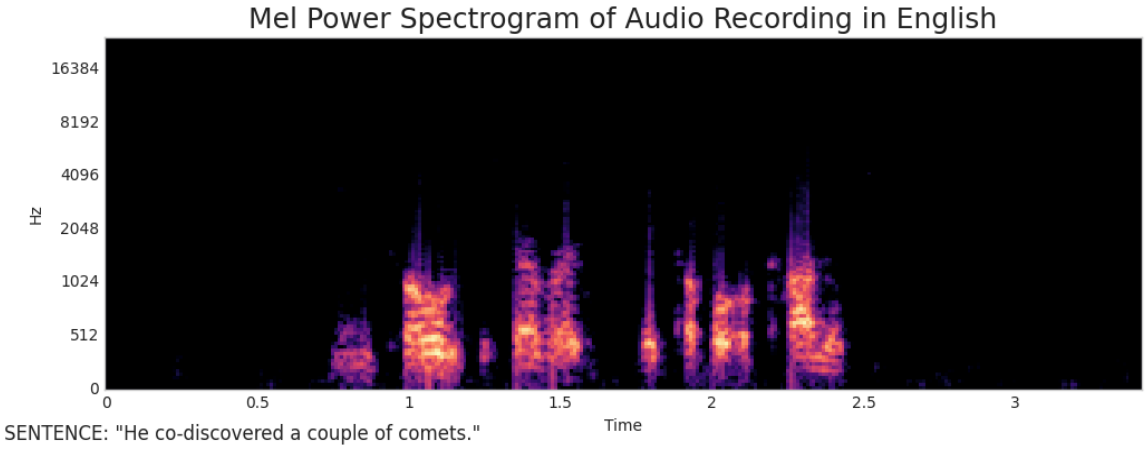


***Figure 3. Mel-Frequency Cepstral Coefficients* Spectrogram Visualization of English Audio Recording**

For the specific model architecture used in this project, the audio data needed to be resampled to 16000kHz and truncated to a maximum duration of 15 seconds in order to be used with the pretrained *Wav2Vec 2.0* model. *Wav2Vec 2.0* is a Transformer-based model used for speech-related tasks like speech recognition and speech representation learning. According to the paper *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations,* "Wav2Vec 2.0 masks the speech input in the latent space and solves a contrastive task defined over a quantization of the latent representations which are jointly learned (Baevski et al., 2020)."

In other words, the model simultaneously encodes useful information into hidden or latent representations of the audio input through masking while also learning to distinguish between correct and incorrect predictions from these representations.The *Wav2Vec 2.0* model was also trained using Connectionist Temporal Classification (CTC) which is meant for sequence prediction tasks where alignment between input and output sequences are unknown, such as in speech recognition.

As mentioned above, audio classification is the categorization of audio signals which can be applied to applications such as music genre identification, emotion recognition, speaker intention, or in the case of our specific project, language identification (Zaman et al., 2023). *Wav2Vec 2.0* offered several advantages for audio classification compared to other Transformer-based models. The model itself was trained using raw audio data from the Librispeech dataset which contains a large multilingual corpus suitable for speech research (Pratap et al., 2020). As a result, it suited well for using the Common Voice dataset, especially the 23 different languages that we selected for our final dataset. In addition, a combination of unsupervised learning techniques and implementation of transformer architecture in pre-training the *Wav2Vec 2.0* model allows it to learn high and low level features and capture contextual information from the input audio data. It is important to note that *Wav2Vec 2.0's* ability to directly analyze raw audio enables it to capture nuanced, language specific acoustic features crucial for distinguishing between languages, making it the superior choice for language classification.

While the main focus lay in developing a language identification model, we also sought to augment the utility of the project by creating an additional transcription model. Since the *Wav2Vec 2.0* transformer is meant for most speech related tasks, it could have been used for

training a transcription model. However, researchers have found OpenAI's *Whisper* transformer model to perform better than *Wav2Vec 2.0* on multilingual recognition and transcription tasks (Kozhirbayev, 2023). *Whisper* is an automatic speech recognition system trained on 680,000 hours of multilingual data from the web (Radford et al., 2023). Unlike *Wav2Vec 2.0*, which leverages unsupervised learning, *Whisper* utilizes a supervised learning approach. Moreover, in contrast to *Wav2Vec 2.0*, which is fine-tuned on a specific dataset, *Whisper* demonstrates exceptional adaptability and makes fewer errors in transcription and translation tasks across diverse datasets due to its training on a wide array of data.

5 Metrics & Results

Due to the extreme size of the Common Voice dataset, our model utilized a manually manipulated version of the dataset resulting in an almost perfectly balanced dataset. Therefore, the metrics used to measure the performance of our model were accuracy and F1 macro score. In the context of our language identification model, accuracy indicates the proportion of correctly labeled languages out of all potential language labels in the dataset. The F1 score was included as an exploratory secondary metric in order to provide a balanced assessment of the models ability to classify languages, taking into account false positives and false negatives. It is important to note that since there is an almost equal distribution of language classes among the dataset, it was expected that the accuracy and F1 score would be nearly identical.

For the *Wav2Vec 2.0* model, we split the dataset into three groups: train, validation, and test. The train set was used to actually train the model, the validation set was used to save the model to the best accuracy, and the test dataset was used to report the final metrics from the model’s performance. This was done to ensure no data leakage occurred when reporting the

metrics of the final model's performance. The train, validation, and test groups comprised 75%, 15%, and 10% of the full dataset, respectively.

Wav2Vec 2.0 Training Process

Considering the complexity of the Transformer model, audio data, and the size of our dataset, the training process took a significantly long time for both of the models. It tended to span from 48 to 72 hours, which significantly reduced the number of models that we had to train. We could only train 1-2 models a week and had to sacrifice some time in order for us to have a selection of models to choose from. Regardless, we were still able to obtain high performing models for our audio classification and transcription tasks.

As previously stated, the dataset was split into three different groups. Train was used to train the model, validation to save to the best accuracy during the training process, and test to get the final metrics of the dataset. We used a test dataset to ensure there was no data leakage when reporting the final metrics of the model's performance. The loss and accuracy of the validation dataset were calculated throughout the training process, which are shown in Figures 4 and 5.

***Figure 4.* Validation Loss in Wav2Vec 2.0**

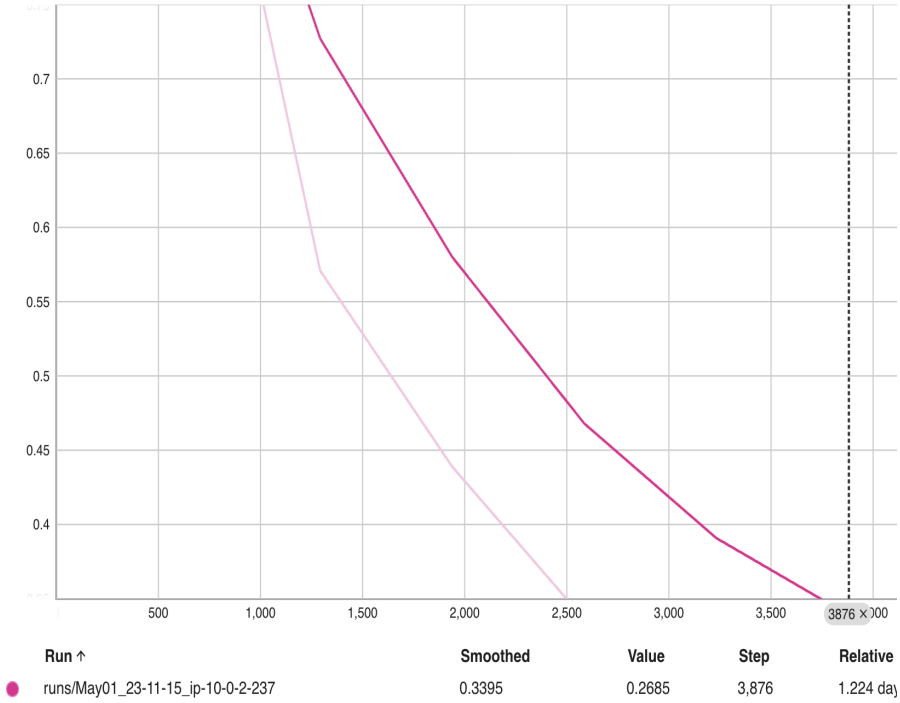
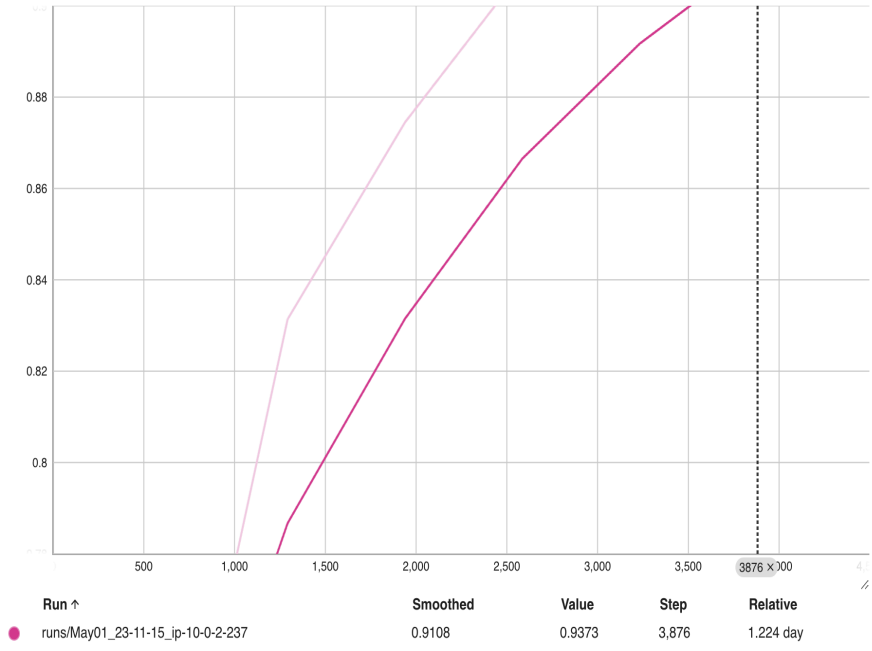


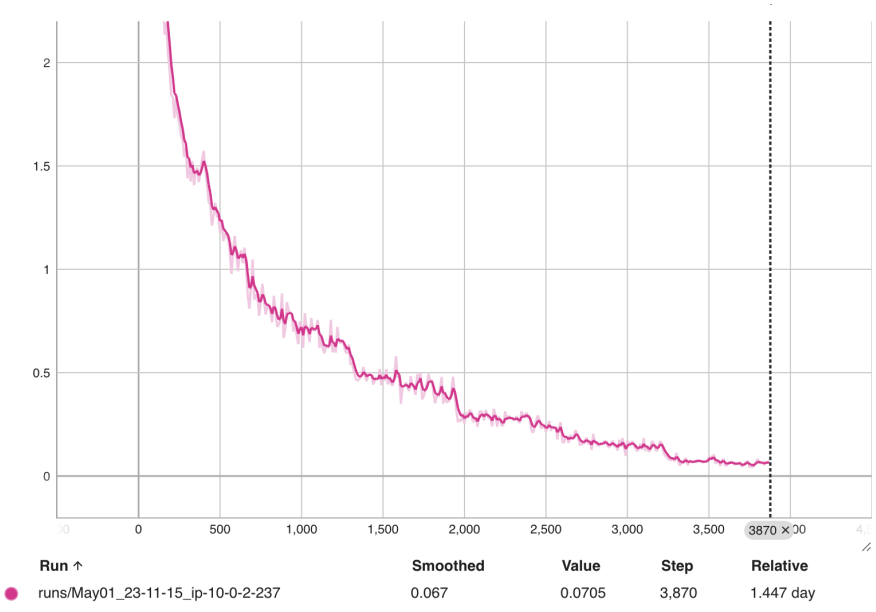
***Figure 5.* Validation Accuracy in Wav2Vec 2.0 Training Process**



As shown above, the loss continually decreased while the accuracy continued to increase. So, the model did not have to revert to a previous step to prevent overfitting. This suggests that the model may have benefitted from further training. However, due to limited time and computational ability, we had to limit the number of epochs to perform for training to six. Regardless, we were still able to achieve good results.

Looking at the decrease in training loss in Figure 6, we notice a steep decline at the beginning, while the loss starts to plateau at the end, signifying it may have started to peak in performance. Regardless, these metrics indicate that we may have benefitted from additional training time or an increased computational ability to speed up the rate of training.

Figure 6. **Training Loss Across Wav2Vec 2.0 Training Steps**



Wav2Vec 2.0 Final Metrics

We were very satisfied with the final performance of the *wav2vec 2.0* model. The final accuracy of the holdout test dataset was 93.61% with an F1 macro score of 93.57. The metrics for each of the three datasets are shown in Table 1.

Table 1. **Final Metrics of Wav2Vec 2.0 Model**

	Accuracy	F1 Macro
Test	93.61%	93.57%
Validation	93.73%	93.72%
Train	99.26%	99.26%

Looking at the performance of each of the languages, we can see good performance across all languages. All 23 of the selected languages had high F1 scores of at least 87. The metrics for each language in the test dataset are shown in Table 2. Certain languages performed better than others. For example, we saw the best performance for Portuguese and Kabyle, while languages like Russian, Catalan, and Esperanto had slightly lower metrics. Despite these differences, the model still had great performance over many different classes.

Table 2. **Metrics for Individual Languages**

	Precision	Recall	F1 Score
English	97	95	96
Catalan	87	94	90
Kinyarwanda	92	96	94
Belarusian	92	94	93
Esperanto	88	92	90
German	91	94	93
French	95	89	92
Kabyle	97	98	98
Spanish	95	92	94
Luganda	97	91	94
Swahili	91	89	90
Farsi	96	94	95
Italian	96	96	96
Meadow Mari	95	97	96
Chinese	92	94	93
Bashkir	89	95	92
Tamlin	91	90	91
Russian	92	82	87
Basque	98	93	95
Thai	92	93	93
Portuguese	98	98	98
Polish	97	98	97
Japanese	96	98	97

Confusion matrices for the three different datasets are shown in Figures 7, 8, and 9. As you can see, the vast majority of predictions are all there datasets are along the diagonal axis indicating correct predictions. There appears to be a few slightly biased incorrect predictions, such as a few russian audio clips being classified as Kinyarwanda. However, these errors are few and far between. Overall, the performance of our model has exceeded our expectations.

Figure 7. **Confusion Matrix for Train Dataset**

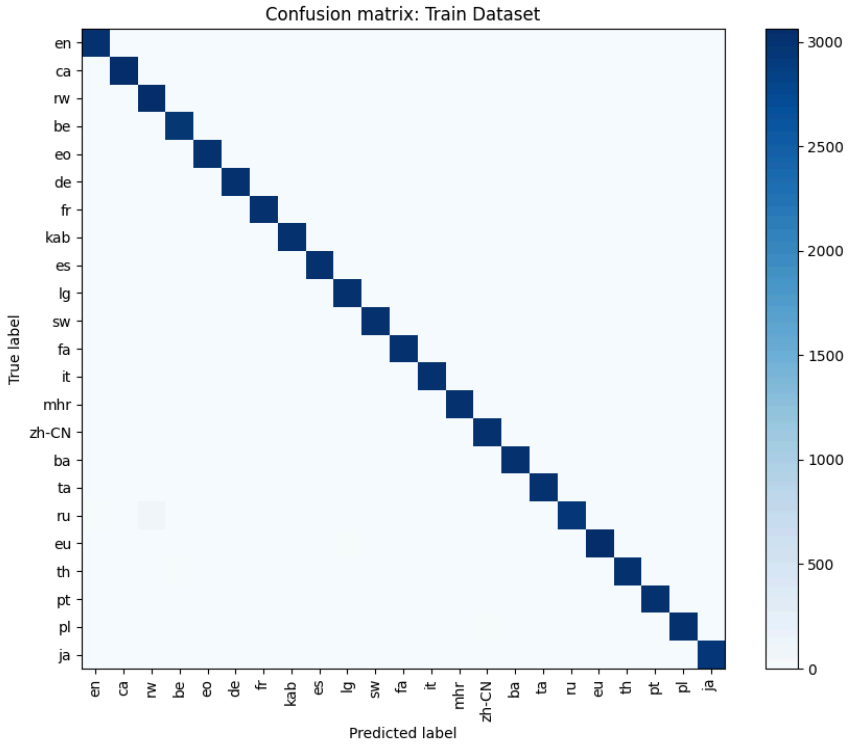


***Figure 8.* Confusion Matrix for Validation Dataset**

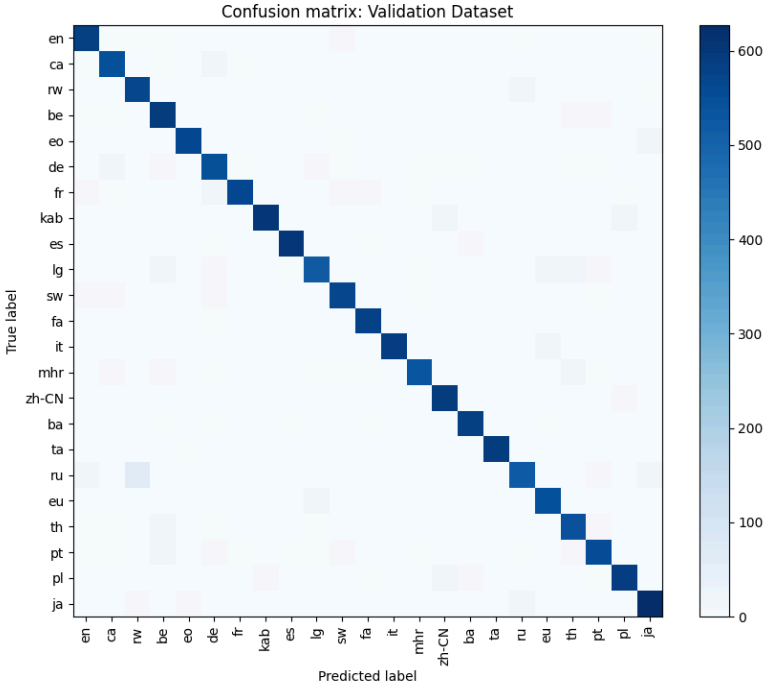
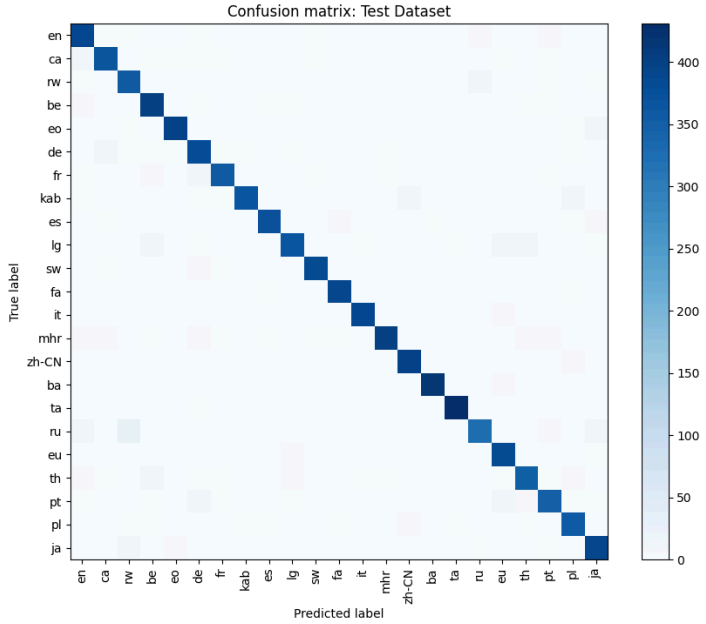


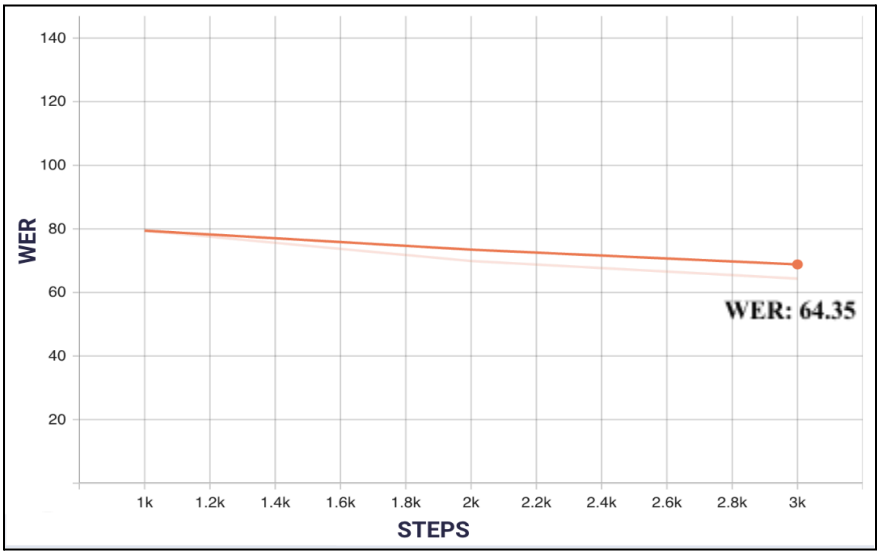
***Figure 9.* Confusion Matrix for Test Dataset**



Whisper Final Metrics

Due to the diverse range of error types (e.g., substitutions, insertions, deletions) and varying lengths of transcriptions generated by the *Whisper* transcription model, accuracy was not the most suitable measure for evaluating its effectiveness. Therefore, WER (Word Error Rate) was used as the main metric to measure its performance. Word Error Rate (WER) is a metric used to evaluate the performance of automatic speech recognition systems by measuring the disparity between the words spoken (ground truth text) and transcribed text generated by the system. It calculates errors in terms of substitutions, deletions, and insertions, providing an idea of how well a transcription model performs (Park et al., 2008). In other words, a lower WER means that there are fewer errors in the transcribed text compared to the reference text or speech. The graph shown below illustrates the decrease in WER over time as the model trained. It's important to note that due to limited computing capacity, the model took several days to train. Therefore, it can be implied that the WER would likely be lower if more time was allotted for training. The model achieved a final WER score of 64.35 indicating the effectiveness of the Whisper transcription model in accurately transcribing speech across the 23 languages used.

Figure 10. **Word Error Rate (WER) Over 3000 Training Steps**



6 Discussion

The root of many of our issues stemmed from our lack of storage and computing capacity. As mentioned previously, audio data is extremely robust, considerably large, and quite taxing on normal computing systems such as the ones used for this project. In an effort to add to the current literature in the field of language identification and deep learning, our model was trained on 23 languages. This approach contrasts other research which mainly focuses on training language identification models using only one language (Kozhirbayev, 2023). However, our goal to utilize multiple training languages through the Mozilla Common Voice dataset posed a significant challenger to our storage capacity, given the intensity of the audio data involved. When accessing through Hugging Face, Common Voice 16.1 is structured in a way where each language is its own dataset within a larger dataset so multiple language subsets cannot be accessed together. The subsets also varied significantly in size. For example, languages like English and Spanish took up almost 250+ GB of storage when loaded onto our local machines. Even if we wanted to take a portion of each language dataset, it was not possible to do so without loading the entire dataset first. Fortunately, we discovered a successful yet tedious solution to create a manually modified dataset without encountering storage issues. This process involved repeatedly loading a single language subset, adding a portion of it to a csv, and clearing the subset from our local machines. The drawback of this solution meant that the dataset lost its HuggingFace qualities and compatibility. It is also assumed that a lot of issues experienced with training and using the Hugging Face interface may have resulted from manipulating the original dataset. Specifically, when we attempted to push the model and training arguments to its HuggingFace repository, our modified version of the Common Voice 16.1 dataset, along with its

metrics and results, were not recognized, despite following the HuggingFace guidelines for fine-tuning a transformer model for language identification.Due to Hugging Face's inability to recognize the dataset and model parameters, the automatically generated audio classification interface yielded contradictory and questionable results despite the model's high accuracy post-training. Luckily, we were able to build our own UI with the models that we trained that were able to make correct predictions.

There are a few other features, analyses, and approaches that would've been beneficial to the model had the storage problems and time commitment to training the model not been so extensive. First off, *Wav2Vec 2.0* was a great transformer architecture given the circumstances surrounding the project. However, prior research has shown that there are other architecture types that would have fared better for our model and overall goals for the project. For example, there are Transformer architecture types that were trained on hundreds of hours of audio recordings from numerous languages but we did not have enough storage or computing capacity to utilize them. Future research should utilize more complex Transformer models which can include more languages and explore other language features like accent or dialect detection.

7 Conclusion

As the computation power of deep learning continues to improve day by day, the practical implications of these advanced models continues to widen. In this project, we demonstrated how powerful these tools can be in regards to audio data and language recognition. By using existing model architecture, we were able to develop models to both predict the language of a speaker from the audio clip, as well as be able to transcribe the script in their spoken language.

With the *Wav2Vec 2.0* model, we were able to successfully build a classification that could predict a language from a short audio clip. This makes this model very practical in the real world, allowing a language to be predicted in a very short period of time, rather than needing a longer clip of audio and thus more information. Additionally, we were able to demonstrate that a transformer model that was originally trained on English, can be applied to 23 different languages. Going forward, it would be interesting to expand to even more languages, considering there are hundreds spoken around the world. However, this was difficult for us to do due to our limited computational power. Regardless, having 23 different languages is still a great breadth of languages for identification. The Whisper model also showed how we can transcribe spoken audio in an accurate and quick way. These two models in conjunction achieved exactly what we had set out to complete: a model that can predict a language with no prior information, and another one to transcribe the spoken text.

As we have discussed, overcoming language barriers has been a huge barrier for centuries, especially with our globalized economy and interconnected world. We were able to demonstrate here the practicality of deep learning to help overcome these barriers. As data scientists continue to advance the development of transformers and other audio related models, language barriers can continue to be overcome. Deep learning can help us bring our communities closer than ever and facilitate conversations on a global scale.

References

Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). Wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems, 33*, 12449-12460.

Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006). Connectionist Temporal Classification: labelling unsegmented sequence data with recurrent neural networks. *Proceedings of the 23rd International Conference on Machine Learning - ICML '06*.

Kozhirbayev, Z. (2023). Kazakh speech recognition: Wav2vec2. 0 vs. whisper. *Journal of Advances in Information Technology, 14*(6), 1382-1389.

Park, Y., Patwardhan, S., Visweswariah, K., & Gates, S. C. (2008). An empirical analysis of word error rate and keyword error rate. Paper presented at the *Interspeech, , 2008* 2070-2073.

Por, E., van Kooten, M., & Sarkovic, V. (2019). Nyquist–Shannon sampling theorem. *Leiden University*, *1*(1), 5.

Pratap, V., Xu, Q., Sriram, A., Synnaeve, G., & Collobert, R. (2020). Mls: A large-scale multilingual dataset for speech research. *arXiv Preprint arXiv:2012.03411,*

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. Paper presented at the *International Conference on Machine Learning,* 28492-28518.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł, & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing* Systems, 30

What is audio classification? - hugging face. *What is Audio Classification?* - Hugging Face. (n.d.-a). https://huggingface.co/tasks/audio-classification

Zaman, K., Sah, M., Direkoglu, C., & Unoki, M. (2023). A survey of audio classification using deep learning. *IEEE Access,*