Vrije Universiteit Amsterdam

Universiteit van Amsterdam

Master Thesis

"Small" language limited-vocabulary automatic speech recognition using Machine Learning

Author: George Vlad Stan 2534186

1st supervisor: Anna Sampaio Bon
2nd reviewer: Hans Akkermans

A thesis submitted in fulfillment of the requirements for the joint UvA-VU Master of Science degree in Computer Science

August 29, 2021

"You want people in AI who have compassion,
who are thinking about social issues,
who are thinking about accessibility"
Timnit Gebru

Abstract

Artificial Intelligence technologies have gained momentum in the past decade, but their usage and research focus still is mainly on the industrialized countries, where big data repositories are available in the cloud, and computing power is massively available. Yet, the applicability of AI for constraint environments, e.g. in poor developing regions of the world, is still under-explored. In this thesis project we explore the applicability of AI for automated speech recognition (ASR) of rare languages, i.e. languages for which no commercial ASR systems exist. We show how a crowdsourcing application for the collection of a simple, small corpus can be effectuated and analysed with AI, as to build voice-based mobile applications for people in low resource environments such as rural Mali and Ghana. As a proof-of-concept, we created a low resource voice data collection application for a multitude of languages and we implemented a machine learning model which uses that small amount of data to understand the words "yes" and "no" in the English language, with up to 98% accuracy. Both applications can be adapted for different words and languages.

I would like to thank all the people who helped and contributed to my work on this thesis, including Anna Sampaio Bon, André Baart, Francis Dittoh, Hans Akkermans, Enrico Rotundo, Alberto Caroli, Nana Kwame Nyame-Essilfie, Charilaos Mulder, Celine Paschal, and Seline Olijdam, as well as everyone who contributed with their voices and translations.

Contents

List of Figures

List of Tables

LIST OF TABLES

1

Introduction

1.1 Context

Artificial Intelligence technologies have gained momentum in the past decade, but their usage has been mainly focused on the industrialized countries, where big data repositories are available in the cloud, and computing power is massively available. Solving societal problems can also be one of the focuses of AI growth, especially in disadvantaged regions of the world. This project aims to do just that, to leverage the strength of AI algorithms in order to aid any number of societal improvement projects around the world, by creating a democratized, free of charge speech recognition system for any language, in any community, no matter how uncommon it is. Its secondary objective is to promote AI as a tool that can be democratic rather than a service only accessible at a high cost from large technology companies.

As an overview of the project, we will start by creating an application for collecting voice data, after which we will gather enough of this data for a specific language, process that data to make it easier for the chosen machine learning algorithm to classify it and create a machine learning model that can successfully and accurately classify each word available.

1.1.1 Software systems for farmers in Sub-Saharan Africa

In developing rural areas in Sub-Saharan Africa, the survival of entire market value chains as well as the communities that rely on them can be greatly aided by providing farmers with important needed information (3), along with collecting information from them about their needs and situation. With this objective in mind the ICT4D team in the Computer Science department at the Vrije Universiteit Amsterdam, together with Web alliance for

Regreening in Africa (W4RA) have been looking for a way to collect and distribute important information to and from farmers in Mali and Ghana through telephone calls. They have created a software system for this use case, called KasaDaka. In previous iterations of this software, the number pad of mobile phones was used, yet this process proved to sometimes be cumbersome and error prone for some users. That is where the need for speech recognition of rare languages appeared. If data can be collected by simply asking yes or no questions or numbers from 0 to 9, the entire information process can be simplified, made more effective and more user friendly. Work for this use case is intended to become a part of a larger project named AfrAiCa, which aims to contextualize AI and IT to the "rural" African context, creating fair ecosystems, and building inclusive platforms through open, grassroots innovation, and local business, without the reliance on big technology companies.

1.1.2 Additional use cases

A machine learning model that can recognize basic words in a specific language can be used in many other situations where information gathering is crucial in aiding a community. In the Philippines, for example, doctors were looking for a solution to remotely collect information from tuberculosis patients, as their treatment was at that time severely affected by the coronavirus pandemic. They could not treat patients in hospitals due to the high risk that SARS-CoV-2 posed to them and due to coronavirus restrictions. Doctors would in this case have been able to use the ASR system created as part of this project to collect health information, keep track of patients' conditions and provide necessary healthcare services remotely, without putting any additional lives at risk.

1.1.3 Concrete examples

The following figures, figure 1.1 and figure 1.2, are concrete examples of voice-based services which can be adapted to use an Automatic Speech Recognition (ASR) system instead of the buttons on a phone, as they are currently implemented. These figures are adapted from (1, p. 16) and (2, p. 94) respectively.
In both examples, the ASR system is shown in red, with arrows pointing towards parts of the software system where speech recognition could be used. In figure 1.1, the question "Buy/Sell/Coop" is also pointed towards, since the questing can also be adapted for a yes or no answer.

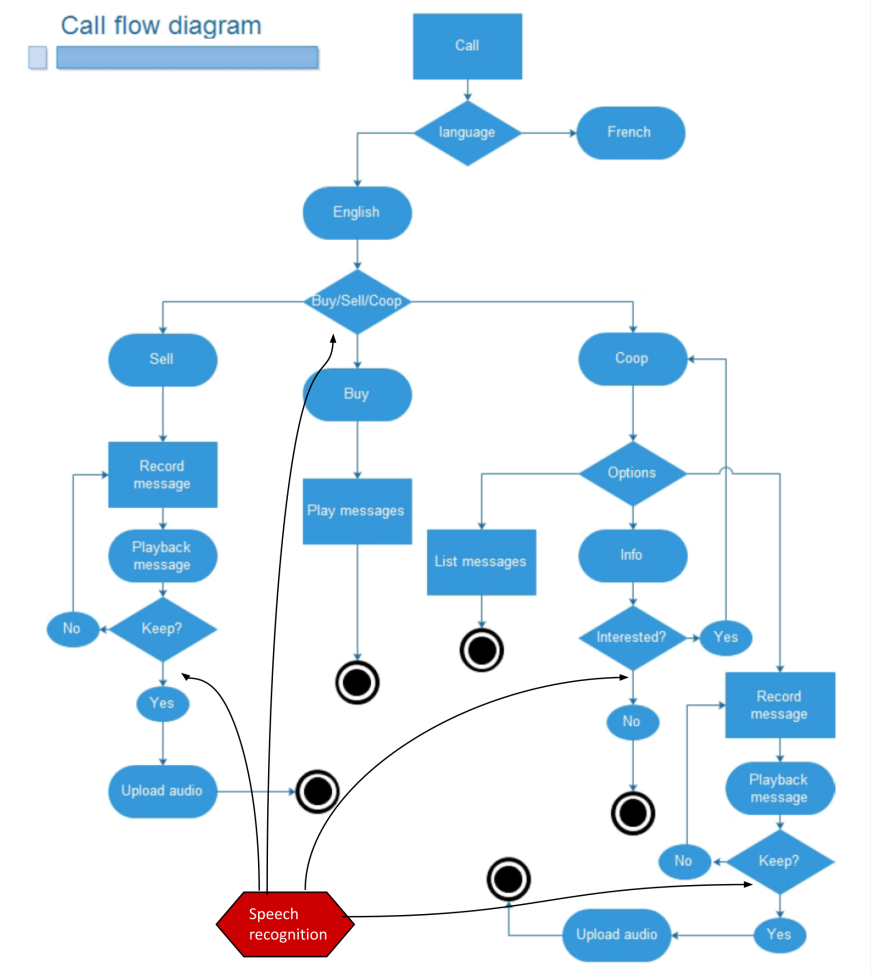

Figure 1.1: Potential ASR adaptation of the Mali Milk Service application. Adapted from (1, p. 16)

1.1.4 Additional motivation

Apart from the aforementioned potential societal improvements possible by building open crowd-sourced AI projects, like the one we are attempting to create, as well as by democratising AI, there are a number of other concerning aspects about the direction the world of artificial intelligence is progressing towards, which serve as additional motivation for our work. As Timnit Gebru's work has shown throughout the years (4), big data language models have an extremely high energy consumption and carbon footprint, due to the vast amount of data models are trained on. This is an especially important aspect to take into account, since marginalised communities are affected worse by climate change (4). Moreover, extremely large amounts of data can be impossible to audit for biases and other

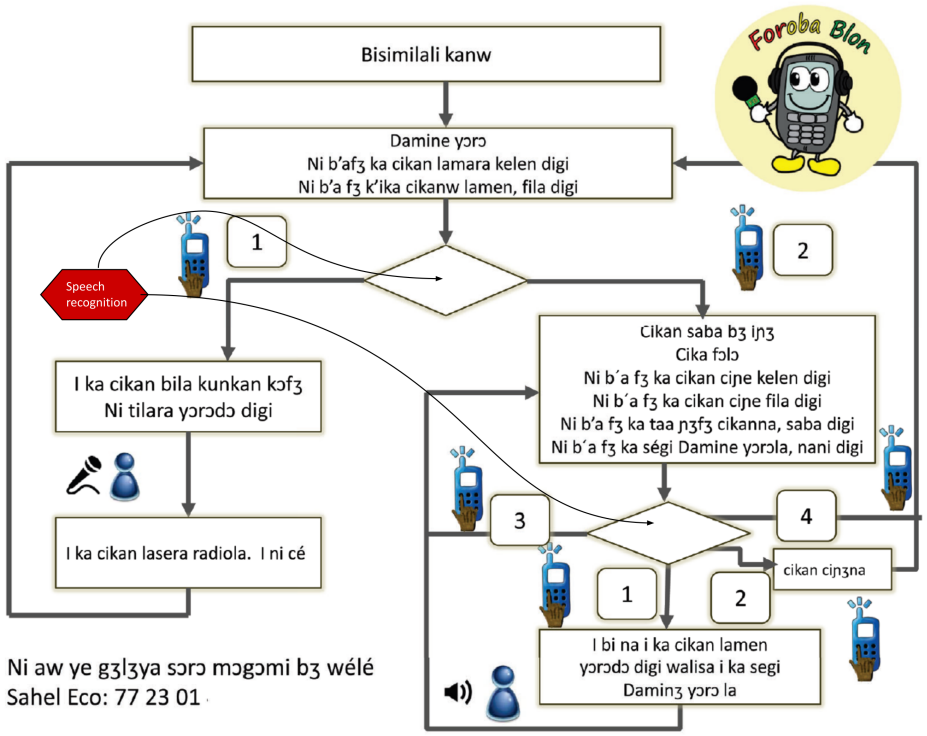

Figure 1.2: Potential ASR adaptation of the call flow model for a mobile voice-based dialogue in the Bambara language. Adapted from (2, p. 94)

undesired characteristics (5). According to her, such large machine learning models are not able "to capture the language and the norms of countries and peoples that have less access to the internet and thus a smaller linguistic footprint online". Without intervening, the result will be that only the practices of the richest countries and communities will be reflected in big machine learning models (4). Machine learning models with limited amounts of crowdsourced data are better suited for tackling the above mentioned concerns.

1.2 Objective

Our first objective is to create a speech recognition system which can recognise the words *"yes"* and *"no"* by processing the speech data and experimenting with different machine learning algorithms to determine the most effective model. Such a system should be able to recognise a variety of different voice types, irrespective of the person's gender, age, or dialect. Expanding this system to recognize numbers from 0 to 9 will also be attempted, depending on the success of the initial functionality. A literature review will be first per-

formed and experts in the field of Artificial Intelligence, specialized in working with audio files, will be interviewed, in order to determine which approach is more likely to be successful in our use case.
An additional objective is to prepare our speech recognition system for integration into KasaDaka, a platform that supports easy creation of local content and voice-based information services, which can be used, for example, to provide information to farmers in a community (6). As mentioned earlier, instead of using the buttons on a phone, which has proven cumbersome in communities where oral communication is preferred, KasaDaka would instead be able to make use of the Automatic Speech Recognition we are attempting to develop.

1.3 Research Question

What is the feasibility of creating low resource application for collecting language vocal data, as well as a machine learning model that can recognize words in sound files with the limited amount of data collected? How many audio data points are necessary to achieve a 90% accuracy in recognising words from audio snippets?

1.4 Research Method

1.4.1 Literature and Feasibility Research

Perform an investigation into the feasibility of the project. Find out what quantity of data is necessary for training and testing a speech recognition machine learning model for simple words. Determine what type of app can be used to collect this data from people on their smartphones, in remote rural areas.

1.4.2 Data collection

Develop an application which can be used to collect voice data in different languages for the desired words that need to be recognized by the model. Start by collecting data for an example language (such as English or Dutch), as an initial test, and then proceed to collect voice data for any desired languages, depending on the use case.

1.4.3 Data processing

Analyse the collected data and determine what further processing is needed for the specific machine learning algorithm attempted and document the findings. The audio data

collected can be converted and processed to any other data type that can fit the machine learning model used.

1.4.4 Machine learning model development

The last part of our process involves attempting to create a model for recognizing the words *"yes"* and *"no"* in either English or Dutch, as a proof-of-concept and in order to discover which potential algorithms have the highest classification accuracy . Using this information we can then move further to developing a machine learning model that can recognize the words *"yes"* and *"no"* in the languages we are interested in, such as Bambara or Twi. Depending on the success of this process, further work can be done towards creating a machine learning model that can recognize the numbers from 0 to 9, which would require starting again from the data collection step.

2

Literature and expert knowledge

2.1 Literature

Before we began our work, it was important to first get an understanding of the context in which our software will be used and the related work that has been done already.

2.1.1 Related work

We learned about the collaboration work done in 2010 in Rwanda and Nigeria towards learning Nigerian Pidgin (7). Similar to our planned approach, the authors converted the audio data to Mel Spectrograms, yet they had a much higher quantity of data available for model training and they attempted to recognise entire spoken sentences.
In papers (8) and (9) the authors investigated which Deep Neural Network acoustic modeling units worked best for developing an ASR system for the languages Chana and Amharic respectively. Similarly, he authors of (10) presented a single ASR model trained on 9 different Indian languages. All three of these works had larger amounts of data available than we predict our project will have.
An open-source tool for collecting speech data, called Woefzela was developed in (11), and was tested with South African languages. This application has the limitation of only functioning on the Android operating system and requiring hardware with high performance, due to the real-time quality control. Its offline functionality would be a good feature for our data collection application.
The authors of (12) performed several experiments in 2010 with data pooling (similar to transfer learning) from related languages in order to improve speech recognition performance. They concluded that as long as the languages are closely related, two hours of speech in a related language is equivalent to one hour of data from the target language, yet

this benefit decreases rapidly if the languages become more distant. In our project, we will also explore the role transfer leaning can play on speech recognition accuracy improvement.
A Text-to-Speech TTS system was presented in (13) which is able to function for the language Twi in Ghana. Many of the limitations encountered by the authors of this paper will also affect our project. Additionally, the language Twi is also one of the target languages for our project.
Another interesting related project we encountered is called Common Voice, an initiative created by the Mozilla Foundation, which aims to crowdsource the data necessary to build both a Text-to-Speech (TTS) system as well as an Automatic Speech Recognition (ASR) system for many different languages, avoiding the reliance on big data companies. A number of our target languages, including Twi and Bambara, are not yet supported by the system.

2.2 Knowledge from experts in the field

I order to get a good understanding of best practices and to determine the up-to-date approach in building a machine learning model for audio classification, four experts in the field of machine learning were consulted:
André Baart, an Artificial Intelligence expert, winner of the Amsterdam Science & Innovation 'High potential Award' with his 'KasaDaka' research project, which provides illiterate West-African farmers with affordable 'internet-like' information services on their simple mobile phones. He is a team member at Bolesian, an artificial intelligence company based in the city of Utrecht, The Netherlands.
Alberto Caroli, a data scientist, working for the company Docebo in Italy, with over 5 years of experience in building machine learning models, with a big part of his expertise in the field of audio classification.
Enrico Rotundo, data scientist with over 4 years of experience building machine learning models, including audio classification, as part of HAL24K, an Amsterdam based company.
Mark Hoogendoorn, a Full Professor of Artificial Intelligence at the Department of Computer Science of the Vrije Universiteit Amsterdam and chair of the Quantitative Data Analytics group. His research focuses on machine learning and its applications, specifically in the domain of health and wellbeing.
Based on their expertise, our literature review, as well as internet tutorials (14), we established that the state-of-the-art method for classifying audio data involves converting the speech sound files to Mel Spectrograms and use them to train a Convolutional Neural

Network adapted to the specific problem, a process which could also function with limited amounts of data. In order to increase the mount of data available as well as to reduce data overfitting, a process called data augmentation can be used, which in our case involves creating new samples by adding artificial noise to the audio files, or changing their pitch. Additionally, as new data is collected after the initial model has been trained, transfer learning can be used to improve the accuracy of the existing model. Furthermore, if other languages exist which are very similar to the target language, data from those languages can be leveraged to improve the model accuracy even further, again using transfer learning.

3

Data collection

3.1 Application

Taking into account the hardware limitations we learned about in the literature review, we decided to start developing an application for collecting the words "yes" and "no" in a list of languages which can be easily expanded and adapted to future needs.

3.1.1 Technology

In order to ensure the application is fit for the limited resource environment of the devices it will likely be used on, we decided to develop our data collection application as a web application, using standard JavaScript, HTML and CSS and avoiding the use of software frameworks which would add significant overhead to the application. The result is a small web application which requires only 490 kB of space on the devices it's running on. In order to record and play back the audio clips, the application makes use of the MediaRecorder interface of the MediaStream Recording standard web API, which is supported by all major desktop and mobile browsers, including Google Chrome, Firefox and Safari. The audio clips collected are compressed using the Opus codec and stored in a Ogg container. In our tryouts, the average size of such a file containing 5 seconds of audio data was just 15 kB, which means it can easily be uploaded even on a very limited bandwidth internet connection (on a 0.1 mbps 2G connection it can be uploaded in less than 2 seconds). The resulting ogg files are then sent to an Amazon S3 bucket. This service can be replaced at any time with a private server or a different cloud storage service. The source code for the application was stored in a private git repository on GitHub (https://github.com/vladpke/rare-language-recorder). The web application was then published using GitHub Pages on the domain https://vocesrares.nl/.

3.1.2 Features

The data collection web application currently supports 18 different languages including Twi, Frafra, Bambara, Mooré, and Bomu. Languages can be added and removed as needed using a simple text editor. The user needs to first give permission for the application to use the microphone after which they are able to see a visualization of the sound wave generated by their voice. They first record the word for "yes", and afterwards the word for "no". The application automatically stops recording 5 seconds after tapping the record button, in order to prevent unnecessarily large files from being uploaded to the server. Before they submit the recording files to us, users are able to play back their recording to make sure they are satisfied with the quality of the recording. If they are not, they can start over with recording both words, until the quality is satisfactory. The final step is to tap the Send button, which uploads both audio files. The web application has a responsive user interface, which means it can be displayed correctly on any type or size of screen. In figure 3.1 one can see the home page of the vocesrares.nl web application with the Twi language selected from the drop down menu.

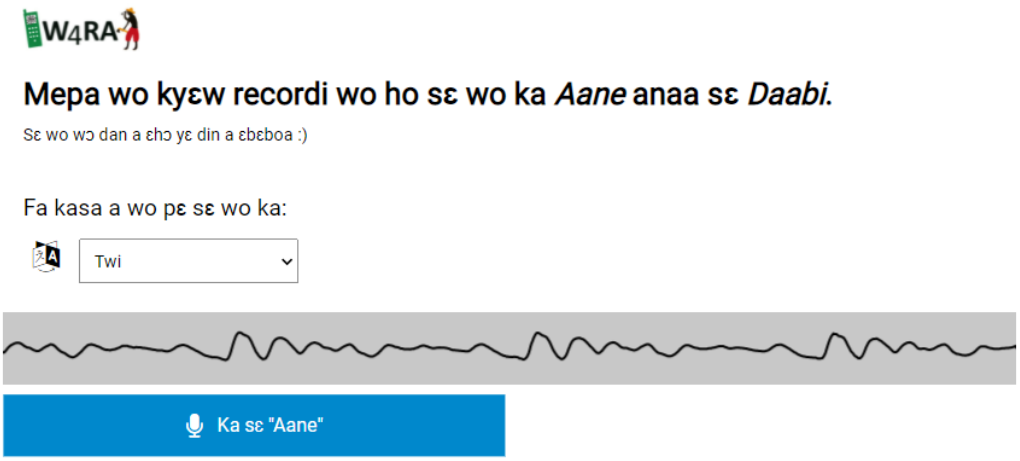

Figure 3.1: Home page with the Twi language selected

3.2 Collection

The application can be used to collect speech data at any time by simply sharing the URL with persons who speak the desired language. During the Open International Webinar **Artificial Intelligence in & for the Global South**, which took place between the 2nd

and 4th of June 2021, as well as the follow-up course at the Vrije Universiteit Amsterdam, a large amount of speech data was collected from the participants, for different languages. English had the most respondents and was chosen as the language that will be used as the proof-of-concept for our automated speech recognition system. A total of 104 speech recordings were collected per word.

4

Data processing

4.1 Analysis

The next step, after collecting sufficient data for a specific language, in our case English, was to analyse the speech recordings collected. In figure 4.1 the wave forms of nine data points along with their respective labels are shown; despite containing the pronunciation of the same word, the wave forms vary a great deal from one another.

4.1.1 Data quality

Each file in our dataset was individually assessed to determine whether the quality of the recording was sufficient for use in the machine learning stage of our project. The results of our analysis showed that 20 files out of 208 were not readily usable for training a machine learning model or were completely unusable. 8 of the unusable files contained only noise or no sound data at all. The remaining 12 files contained the right words pronounced by different individuals, but repeated multiple times within the same file. Our decision was to split these multiple utterances into separate files, increasing the amount of data we had available. Our final dataset contained 248 audio files, 124 with the word "yes" and 124 with the word "no".

4.1.2 Data compatibility

An obstacle we encountered from our first attempt to process the data was the format incompatibility with certain Python libraries. We noticed that depending on the browser that the recording was made in, the codec used to encode the audio data was different, which resulted in errors when using certain files as inputs. We circumvented this problem

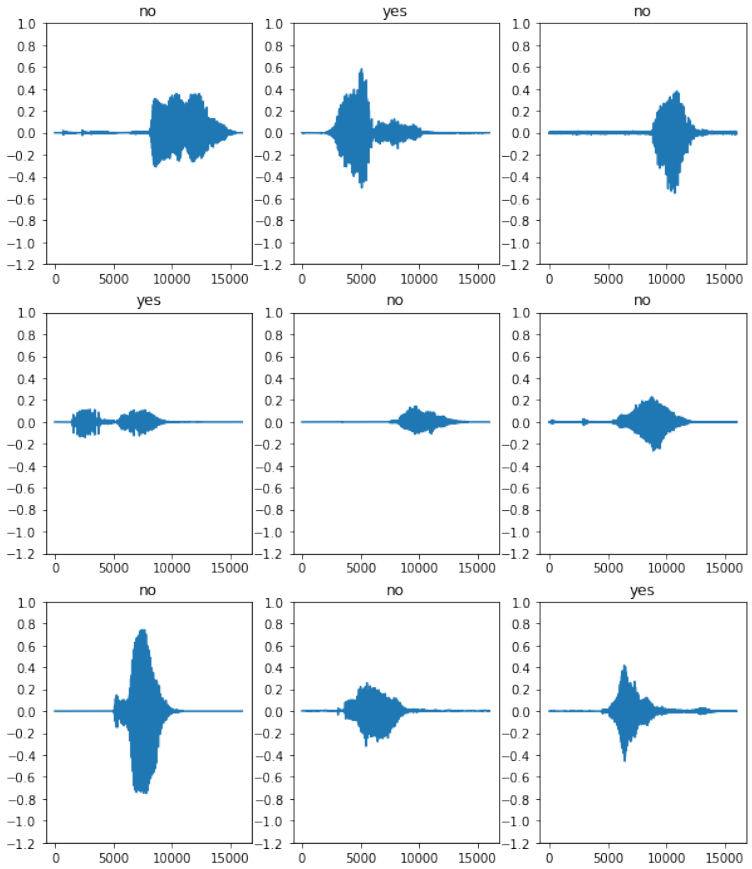

Figure 4.1: Wave signals of different data points

by adding different input procedures for different encodings, and by converting all the files to the wav file format.

4.2 Preparation

Our next step involved preparing the data for use in a Convolutional Neural Network machine learning model.

4.2.1 Data augmentation

Data augmentation is a process in which certain techniques are used to increase the amount of data by adding slightly modified copies of already existing data. In our case this process involved taking the existing sound files and modifying their pitch or adding artificial noise, or both. For each of our audio files we created two new versions of the file, one with lower pitch and one with higher pitch. We then took the three resulting files and added artificial

noise to them, resulting in a total of six files. This meant that using data augmentation, we now had six times more data available, which amounted to 1488 audio files. The data augmentation process not only increases the amount of data, but also helps increase the accuracy of our speech recognition by preventing the machine learning model from overfitting during training and by making the model's predictions compatible with different voices and noisy environments.

4.2.2 Standardization

At this point in our project, the dataset we had collected included audio files of different lengths and of different encodings. The machine learning model we planned on using expected input data of the same size and of the same format. Because of this we decided to standardise all our data to be 1 second in length and to use a 16 kHz 16-bit single channel PCM wave file format. We also normalized all our audio data to a standard amplitude.

4.2.3 Conversion to Mel Spectrogram

In order to prepare our audio dataset for the training stage of our convolutional neural network, we must first transform the sound signal into a mel spectrogram, a two-dimensional visual representation of the spectrum of frequencies of a sound signal as it varies with time. Mel here refers to the use of the Mel scale, which is a non-linear frequency scale in which sounds of equal distance from each other on the graph also sound as being equal in distance from one another to humans. For example, in the hertz (Hz) scale, the difference between 500 and 1000 Hz is obvious to a person, whereas the difference between 8000 and 8500 Hz is not noticeable.

In order to transform each audio signal into a spectrogram, we had to first compute the short-time Fourier transform for each file, which is a mathematical function that gets a signal in the time domain as input, and outputs its decomposition into frequencies. Since our files are all 1 second in length, we used a short-time Fourier transform with window-size of 255 and a hop-size of 128. These parameters can be adjusted and determine the resolution of the resulting spectrogram. Next we computed the sound magnitudes at each frequency window, in decibels (dB) and we converted the linear hertz scale to the Mel scale mentioned above. We now had our spectrograms ready to be used in the training of the machine learning model. In figure 4.2 a number of spectrograms are shown for random files in our dataset, along with their corresponding labels, "yes" or "no".

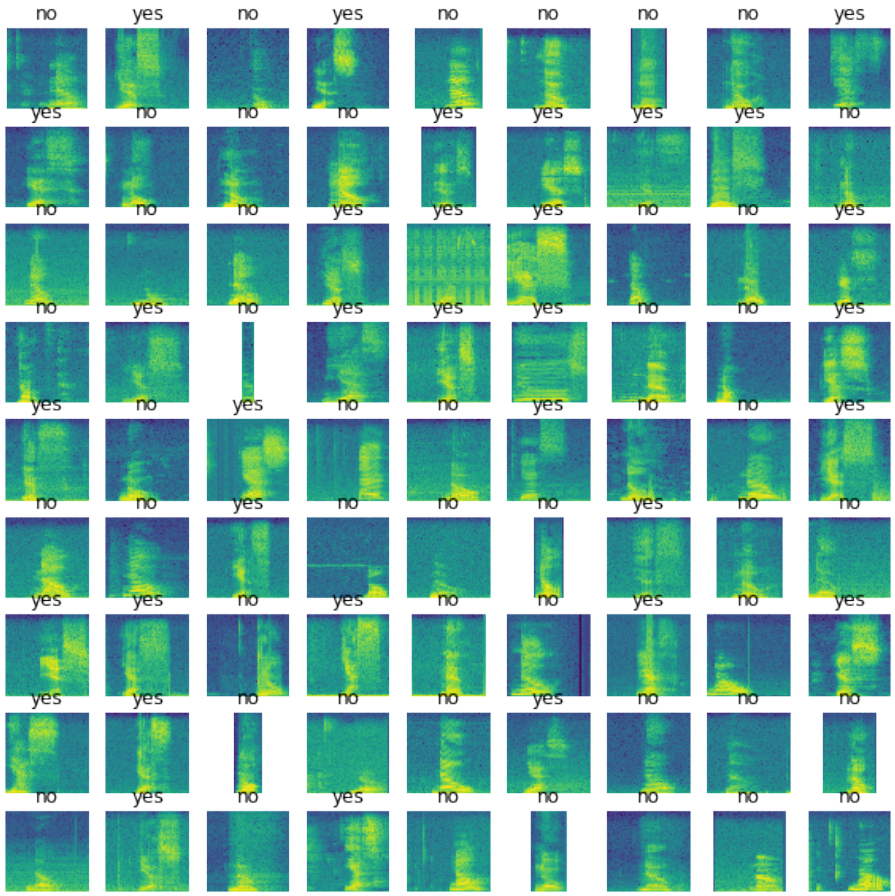

Figure 4.2: Visualization of mel spectrograms of random files in our dataset

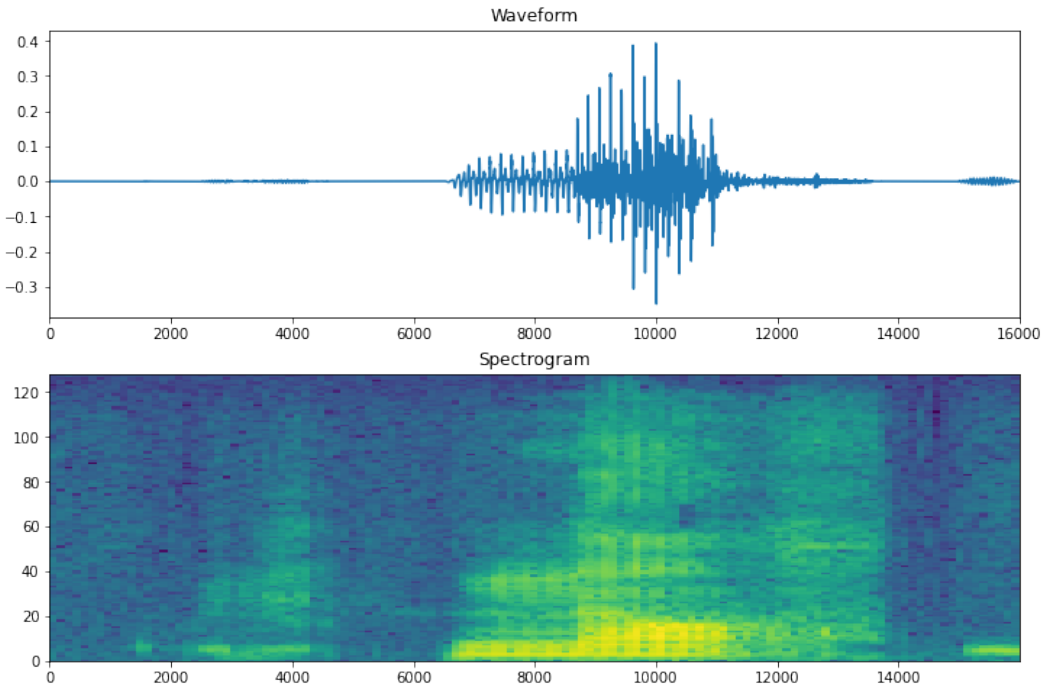

Figure 4.3: Comparison of a wave signal with its corresponding spectrogram

5

Machine learning model

5.1 Training

In this chapter we will talk about the construction of the machine learning model using the data we prepared in the previous chapter for training, validation and testing. We first randomized the data after which split it into three datasets using the percentage ratio 70:20:10. That means 70% of the data was reserved for the training dataset, 20% of the data for the validation dataset, and 10% of the data for the testing dataset. Other ratios were attempted, such as the 80:10:10 ratio, but they yielded poorer accuracies. There are a number of Python libraries that can be used to create a convolutional neural network machine learning model. The two libraries that we tried working with were Keras, an open-source library which is now part of the TensorFlow library, and fast.ai, which is a deep learning library built by a non-profit research group. Our final yes/no model was created using the Keras library, due to the familiarity with the syntax and the high availability of technical knowledge support on the web. For our project, the library can be swapped at a later time, if deemed necessary. Using the Sequential model from the Keras library, we were able to build our convolutional neural network. This type of neural network contains convolutional layers, based on the mathematical operation of convolution; they are sets of filters, in the shape of 2D matrices, convolved with the input image during learning, enhancing distinguishing features in it and aiding greatly in computer image classification. Figure 5.1 illustrates the entire configuration of our CNN model (15).

In our final model configuration we made use of:

- a Resizing layer to downsample the input, enabling the model to train faster

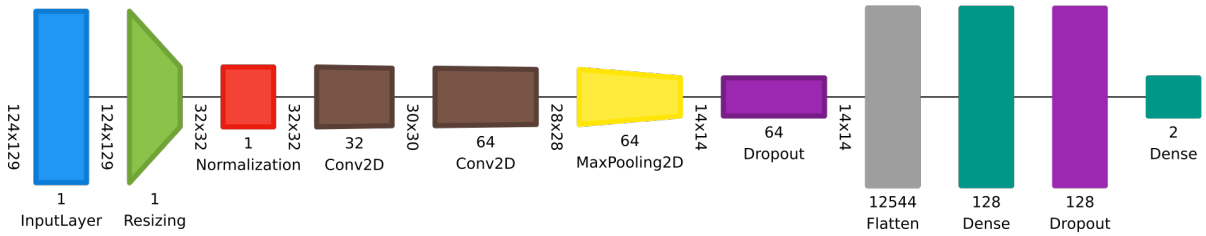

Figure 5.1: Model architecture visualization

- a Normalization layer to normalize each pixel in the image based on its mean and standard deviation
- two Convolutional layers with 32 and 64 output filters respectively, both using the Rectified Linear Unit (ReLU) activation function and a 3x3 kernel
- a two dimensional Max-pooling layer, which down samples the input in order to highlight the most present feature in an image or output matrix
- a Dropout layer which randomly sets input units to 0 with a desired frequency, which in our case is 0.25; this layer reduces over-fitting
- a Flatten layer, which converts the data into a 1-dimensional array for inputting it to the next layer
- a Dense layer, which is a neural network layer that is connected deeply, which means each neuron in the dense layer receives input from all neurons of its previous layer; the Dense layer we used has an output dimensionality of 128 and uses the ReLU activation function
- an additional Dropout layer with a 0.5 probability setting
- finally, a second Dense neural network layer with an output dimensionality of 2, for the number of labels we have in our data, "yes" and "no"

The model was then compiled using the Adam optimization algorithm.

5.2 Testing

Our next step was to test the resulting machine learning model with the testing dataset as well as with real life speech inputs. For each combination of parameters and quantity

of data attempted we followed these steps 10 times and computed the average accuracy: re-randomised the data, trained a new model and collected the resulting accuracy. In table 5.1, the resulting accuracies are shown depending on the total quantity of data used.

2000 samples	97% - 98% accuracy
400 samples	90% accuracy
200 samples	84% accuracy

Table 5.1: Data quantity impact on model accuracy

The confusion matrix in figure 5.2 shows a comparison between the word predicted by the model and the actual label the sound file from the testing dataset had. When the model predicted the word "no", it was correct 100 times and wrong 6 times, while when the model predicted "yes" it was correct 95 times and wrong 7 times. The accuracy of the model needs to be extensively tested further in real life. In our limited real-life tests, the accuracy was very high.

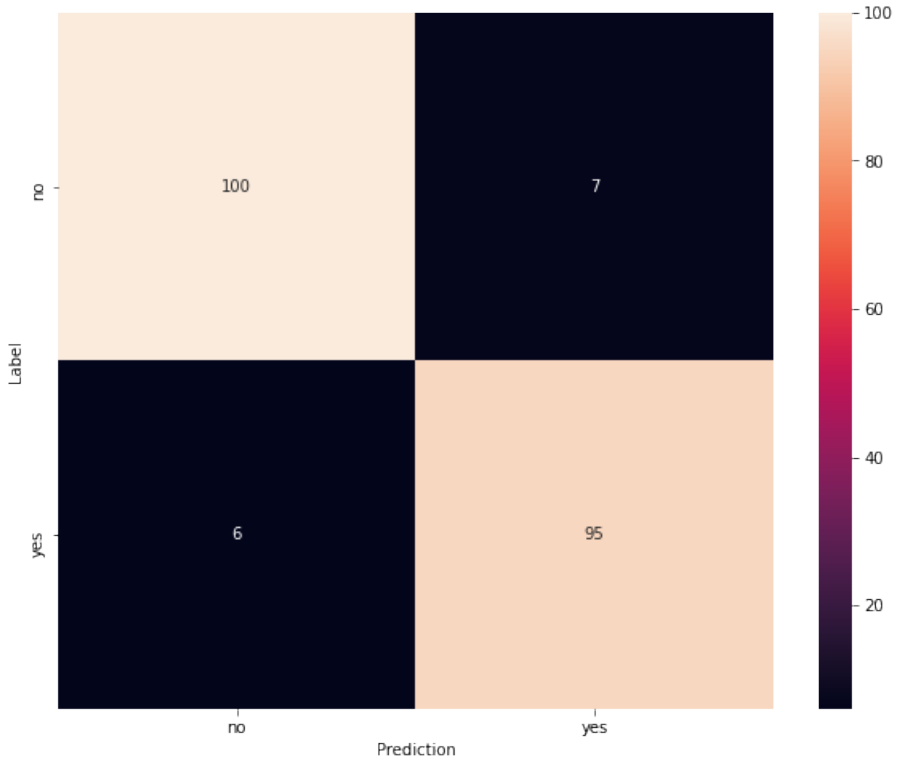

Figure 5.2: Confusion matrix after model testing

5.3 Model deployment

In order to showcase the functionality of our yes and no speech recognition model, we built a Flask web application which uses voice control for interaction and provides farming related information. The model is integrated in the web application and can be accessed by sending it an audio clip, after which it returns its prediction. This application also demonstrates the ease with which the model can be deployed in any type of application, online and offline. Figure 5.3 is a screenshot of the Flask application incorporating the machine learning model.

Figure 5.3: Flask web application using the machine learning model

6

Future work and conclusion

6.1 Future work

The results of our project highlighted just how much future work is possible and necessary to develop and improve our limited ASR system and to make sure it actually ends up helping communities around the world which need it the most.

Our ASR system could be expanded to support a larger vocabulary, such as numbers from 0 to 9, as well as more languages which have no ASR support currently. We performed experiments with number recognition in addition to the words "yes" and "no" and the results were promising.

The multiple biases introduced by our data can be mitigated. A good range of voices needs to be supported which does not favor a specific gender, age or dialect. For example, if the data collected comes predominantly from males between the ages of 30 and 50, the model will have a disproportionately better accuracy for individuals in that group. Similarly, if the data contains only speakers of a certain dialect of the target language, the model will perform better with speakers of that dialect.

During our data collection phase, we noticed that some of the recordings were done either too far or too close to the microphone, which resulted in audio data that couldn't be recognised even by human listeners. It is important that in the future, during data collection, we encourage users to listen back to their recordings before they submit them, and re-record themselves if necessary. If they are unable to record a clear enough sample, likely due to a faulty microphone, we should encourage them to not submit their recordings and instead try a different device.

Our data collection application currently only interacts with users via written text. Considering the predominantly oral communication present in many of the targeted communities,

a version of our application should be created that works with vocal instructions.
As the amount of data for our model training increases, so does the difficulty in detecting flaws in data samples. An automated system should be developed which can flag audio files which might not fulfil quality standards.
More work can be done in finding better parameters for the Convolutional Neural Network model used, as well as in experimenting with other machine learning models which could yield better performance. The software applications built as part of this project could be made open source and user-friendly so that anyone who desires to build a not-for-profit Automatic Speech Recognition system for their language is able to. Since the currently built model managed to achieve a very high accuracy, it should now be trained with our currently targeted languages, Twi and Bambara, for which data needs to be collected on the field or through the internet. Those models can then be integrated with the KasaDaka (6) software system, in order to add a new way for users to interact with it.
Finally, the models for each of the languages we support should be able to improve over time, using transfer learning with new speech data, as well as data from very similar languages.

6.2 Conclusion

Artificial Intelligence technologies do not need to be centralized in industrialized countries and big companies. There is a high potential for democratizing AI, crowdsourcing its training data and leveraging its power for all communities around the world, no matter their socioeconomic status.
We have manged to build three software applications as part of our project, one for data collection, one for machine learning model training and one for showcasing the accuracy of the resulting model. We have demonstrated that it is feasible to build a democratized, free of charge Automatic Speech Recognition system with a limited vocabulary and using a very low amount of data. With just 200 data samples per word, we were able to achieve an accuracy above 90% for recognising the words "yes" and "no". We have identified many areas in which such a system could be used to improve the quality of services provided by different organizations in communities around the world, and we have set the goal to deploy our ASR software as part of the KasaDaka software system. With more involvement in such projects by data scientist from any part of the world it will be possible to make Artificial Intelligence a tool available for everyone.

References

[1] Aske Robenhagen and Bart Aubers. **The Mali Milk Service 3.0**, 2016. Vrije Universiteit Amsterdam. iv, 2, 3

[2] Anna Bon. *Intervention or Collaboration?: Redesigning Information and Communication Technologies for Development*. PhD thesis, Maastricht University, December 2020. iv, 2, 4

[3] N.B. Gyan. **The Web, Speech Technologies and Rural Development in West Africa An ICT4D Approach**, 2016. Exacte Wetenschappen Naam instelling promotie: Vrije Universiteit Amsterdam Naam instelling onderzoek: Vrije Universiteit Amsterdam. 1

[4] Karen Hao. **We read the paper that forced Timnit Gebru out of Google. Here's what it says.**, 05 2021. 3, 4

[5] Jasmina Šopova. **Audrey Azoulay: Making the most of artificial intelligence**, 08 2018. 4

[6] André Baart. **KasaDaka: a sustainable voice-service platform**. Technical report, Master Thesis Vrije Universiteit Amsterdam, 2017. 5, 23

[7] Daniel Ajisafe, Oluwabukola Adegboro, Esther Oduntan, and Tayo Arulogun. **Towards End-to-End Training of Automatic Speech Recognition for Nigerian Pidgin**, 2020. 7

[8] Tessfu Fantaye, Junqing Yu, and Tulu Hailu. **Investigation of Automatic Speech Recognition Systems via the Multilingual Deep Neural Network Modeling Methods for a Very Low-Resource Language, Chaha**. *Journal of Signal and Information Processing*, **11**:1–21, 01 2020. 7

[9] Tessfu Geteye Fantaye, Junqing Yu, and Tulu Tilahun Hailu. **Investigation of Various Hybrid Acoustic Modeling Units via a Multitask Learning and Deep Neural Network Technique for LVCSR of the Low-Resource Language, Amharic**. *IEEE Access*, **7**:105593–105608, 2019. 7

[10] Shubham Toshniwal, Tara N. Sainath, Ron J. Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, and Kanishka Rao. **Multilingual Speech Recognition With A Single End-To-End Model**, 2018. 7

[11] Nic J De Vries, Jaco Badenhorst, Marelie H Davel, Etienne Barnard, and Alta De Waal. **Woefzela-an open-source platform for ASR data collection in the developing world**. Conference paper, 2011. 7

[12] Charl Van Heerden, Neil Kleynhans, Etienne Barnard, and Marelie Davel. **Pooling ASR data for closely related languages**. 2010. 7

[13] Justyna Kleczar. **General purpose methodology and tooling for Text-to-Speech support in voice services for under-resourced languages**. Technical report, MA thesis. Vrije Universiteit Amsterdam, 2017. 8

[14] Adam Geitgey. **Machine Learning is Fun Part 6: How to do Speech Recognition with Deep Learning**, 09 2020. 8

[15] Alex Bauerle, Christian van Onzenoodt, and Timo Ropinski. **Net2Vis – A Visual Grammar for Automatically Generating Publication-Tailored CNN Architecture Visualizations**. *IEEE Transactions on Visualization and Computer Graphics*, **27**(6):2980–2991, Jun 2021. 18