Speaker ID and Few Shot Keyword Spotting for Edge Devices

By

Rachael Burris

Samantha Patil

Akshay Ratnawat

Siddarth Balasubramani

Supervisor: Utku Pamuksuz

A Capstone Project

Submitted to the University of Chicago in partial fulfillment

of the requirements for the degree of

Master of Science in Analytics

Physical Sciences Division

March, 2022

Abstract

Demand for portable computers, known as edge computing devices, has been on the rise in the past decade. To keep pace with the ever increasing speed of the internet, research into the optimization of edge devices is crucial. Each year, Consumers expect more computing power, faster results, and increased security than previously available. Maxim Integrated, an edge device and computing company, has been working towards developing a cutting edge solution, using embedded technology that conducts deep convolutional neural network operations on low-bit and low-power accelerators (MAX78000). We aim to provide a solution to train a deep learning system that can fit MAX78000 to recognize customized keywords and speaker efficiency by training the Siamese neural network that performs with maximum precision and fewer false alarms per hour of speech.

Keywords: artificial intelligence, deep learning, few shot keyword spotting, neural networks, siamese networks, Google Speech Commands, VoxCeleb

Executive Summary

With the tremendous increase in the number of IoT-connected devices worldwide, there has been a rise in the use of machine learning and artificial intelligence to decode the sensor data to make decisions and predictions. This has put Edge Computing devices on a rigorous development and optimization route in terms of computational efficiency, latency, hardware and algorithmic processing power.

Speaker ID and keyword spotting are techniques developed for the enhancement of such edge devices, however they consume much battery energy during the always on operation. One potential solution has been developed by Maxim, which has an embedded technology that conducts deep convolutional neural network operations on low-bit and low-power accelerators (MAX78000).

This paper aims at providing a solution to train a deep learning system that can fit MAX78000 to recognize customized keywords and speaker efficiency by training the Siamese neural network. The research uses Google Speech Commands dataset to develop the model and user-defined keyword for testing of keyword spotting and CelebVox and LibriSpeech datasets for Speaker ID. The research aims at training the **S**iamese neural network model to learn good representation of features in order to reuse the model's features for verification of speaker ID without retraining any new speakers. Finally, the paper aims to explain the output of the SID and KWS system developed to combine a speaker-aware KWS system that performs with maximum precision and less false alarms per hour of speech observed.

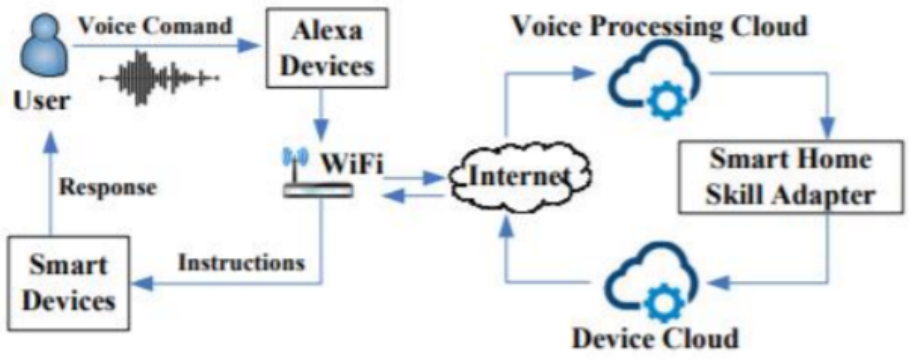
Table of Contents

List of Figures

Introduction

Business at Maxim Integrated is focused on solving engineering problems to empower design innovation and create products that shape the world. One such application of this is wearable technology, or edge devices, where they aim to enhance portable devices by reducing size while increasing in power efficiency and sophistication. Edge devices serve a wide range of uses, though today's interest in hands-free technology and automation make audio recognition a front runner.

Figure 1. *Data flow diagram for common cloud computing devices.*



(Figure1: Tanui, Meshack, 2020)

Problem Statement

Current standards in a popular subset of audio recognition known as speech recognition models present several barriers in their industry. The most common of which is the sheer volume

requirement of data needed in order to attain acceptable levels of accuracy, often measured by the number of incorrect recognitions per alarm (known as false alarm rate). The most popular of such devices in this industry (such as Alexa, Google Assistant, and Siri) achieve low false alarm rates using computationally expensive models that require most small devices to send data to larger remote computers for executing, and sending the response back to the device, introducing two additional challenges – high computation power and internet connectivity. Indirectly, the use of cloud computation introduces security vulnerabilities (Tanui, Meshack, 2020).

Analysis Goals

To overcome these challenges, we propose an artificial neural network for effective keyword and speaker recognition, leveraging unsupervised feature extraction, feature reduction, and distance metric learning methods, to suit Maxim Integrated's low-bit and low-powered edge device using local computation. By leveraging local computation, we veer from the industry standard of cloud computation, providing increased data security and ownership. Further different from previous work, we'll optimize our model to perform on fewer samples of audio, known as few shot learning, reducing the amount of data needed to achieve a high accuracy. An effective audio recognition model can correctly identify who a speaker is as well as what they said. With this two part goal in mind, we'll minimize the false alarm error on two separate models: Speaker Identification and Keyword spotting.

Speaker ID (SID) is a biometric verification technique in which the device is required to correctly recognize the identity of the speaker. In speaker identification, an utterance from an unknown speaker is analyzed and compared with speech models of known speakers. The

unknown speaker is identified as the one whose model best matches the input utterance. If the match is good enough—that is, above a threshold—the claim is accepted (Furui, Sadaoki, 2010) .

Keyword spotting (KWS) is a technique for recognizing certain spoken keywords, such as “Okay, Google” and “Hey Siri”. State-of-the-art SID and KWS systems are built using deep learning systems such as recurrent neural networks. But deploying these deep learning systems on the edge devices require a lot of memory and battery power to process the data locally. Maxim Integrated hardware aims to provide a wearable device with local computation as an alternative to the cloud computing devices that are standard today.

By maximizing the trade-off of physical constraints of computing power and model accuracy, we deliver powerful audio recognition solutions to be implemented on Maxim Integrated’s specialized hardware.

Variables and Scope

Key variables to be investigated therein include feature extraction methods, training sample size, and other neural network features that impact computational expense (GPU and RAM consumption) such as loss function. While we will be optimizing our models for the specific hardware provided by Maxim, as defined in this paper, we will not venture into hardware as variable, as it is outside the scope of this project.

Background

Maxim Integrated is an edge-computing technology company that manufactures and sells a wide variety of high-performance analog and mixed-signal products and technologies. Maxim specializes in solving design challenges related to power efficiency for a wide range of machines

from cars to wearable devices, miniaturization, and security on application-specific solutions in the automotive, manufacturing, healthcare, communications, and cloud computing industries.

Literature Review

Few Shot Learning

Awasthi et al. (2021) recognized the importance for KWS models to adapt to new classes using fewer samples, as providing extensive samples (for instance, over a 1000) is burdensome to users. They go on to implement a model requiring fewer samples (known as few shot KWS) to allow for on device customization and introduction of new words. The model's performance is most improved by implementing a loss function.

Local Computation

Edge computing devices can ameliorate data privacy concerns in that they eliminate the risk of data leakages posed by moving data to the cloud (Mark Ryan 2010). Edge computing also reduces latency down to milliseconds while minimizing network bandwidth (Plastiras, George and Terzi 2018). Still, the most important problem is power saving so that these devices can work for days without charging. Any unnecessary data transfers, storage, or processing represents a waste of energy (Merenda and Massimo 2020).

From the modeling perspective, the efficiency of the edge of computing devices can be enhanced by improving the model design and compression. Designing models with a reduced number of parameters (layers and neurons) can help reduce the memory and execution latency without sacrificing model accuracy. Several models have been optimized for deployment on edge devices, including Mobile Nets (Howard and Andrew 2017), Squeeze Nets (Iandola, Forrest and

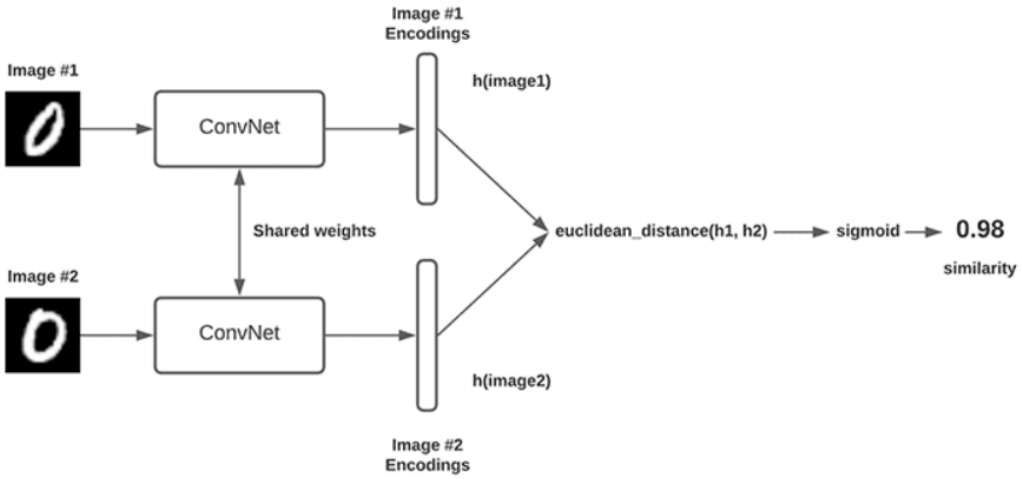
Han 2016). It's important to have a smaller and efficient network to run the model on mobile devices - which has smaller memory and processing power.

Siamese Networks

The Siamese Neural network was introduced for the first time in the 1990s by Bromley and LeCun to solve signature verification as an image matching problem (Bromley and LeCun 1993). A siamese neural network consists of twin networks which accept distinct pairwise inputs but are joined by an energy function at the top. This function computes some metric between the highest-level feature representation on each side. The parameters between the twin networks are tied and shared with each other.

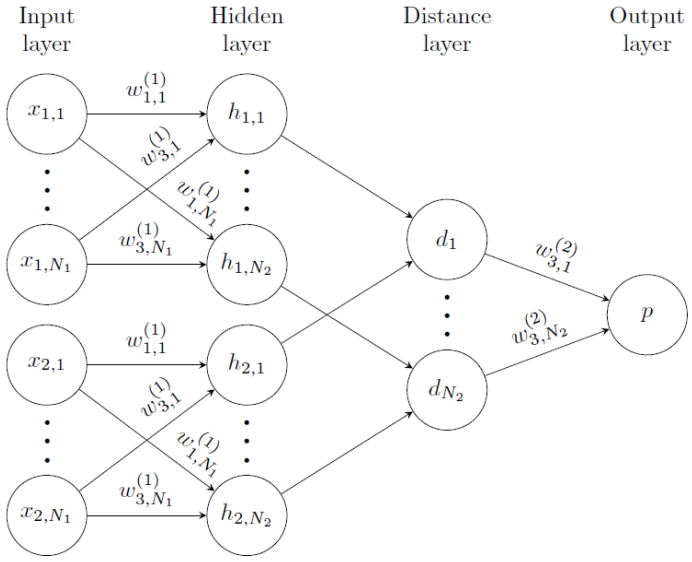
Two key properties of the model, identical parallel layers structure and weight sharing, ensuring the consistency in the predictions. As such, the identical networks could not map two similar data points to very different locations in the feature space because each network uses the same weights and functions. If the audio is similar they should output the same results and if they are different they will output different results. This difference can be captured using different similarity metrics.

Figure 2. *Visual representation of a Siamese Neural Network framework.*



(Source: https://www.pyimagesearch.com/2020/11/30/siamese-networks-with-keras-tensorflow-and-deep-learning/)

Figure 3. *Layer configuration example of Neural Networks.*



(Source: Koch and Zemel, 2015)

Distance Metric Learning

Killian Weinberg and Laurence Saul (2009) noted that traditional similarity models, like k nearest neighbors, rely heavily on the distance calculated between classes. In exploring several methods to exploit this margin, different outcomes can be obtained. The authors go on to posit that the ideal distance between labeled groups can be learned, rather than specified in advance, allowing the model to adapt and more accurately classify different labels. This concept proves to be an effective method for margin maximization and can improve model performance by increasing the separation between classes.

Data

Data Sources

To accomplish both KWS and SID, training data will be collected from multiple sources: Google, VoxCeleb and LibriSpeech. Each data source will provide two features: an mp3 clip (unstructured) and a text label (categorical). Google's Speech Command Dataset will be leveraged for the initial keyword spotting. The dataset is composed of 65,000 one second long clips of 30 short words by thousands of different people via Google's Do-it-Yourself Artificial Intelligence (AIY) website. The dataset is approximately 8.17 Gigabytes. Each label has both a numerical value and text description. For Speaker ID, training will occur on VoxCeleb2, a dataset of over 6,000 three second segments of 1 million utterances. Speech was extracted from interview videos uploaded to Youtube. LibriSpeech will be the final source of data training to improve performance in real world environments with background noise. LibriSpeech is a corpus of approximately 1000 hours of 16kHz read English speech. The data is derived from read audiobooks from the LibriVox project, and has been carefully segmented and aligned. This

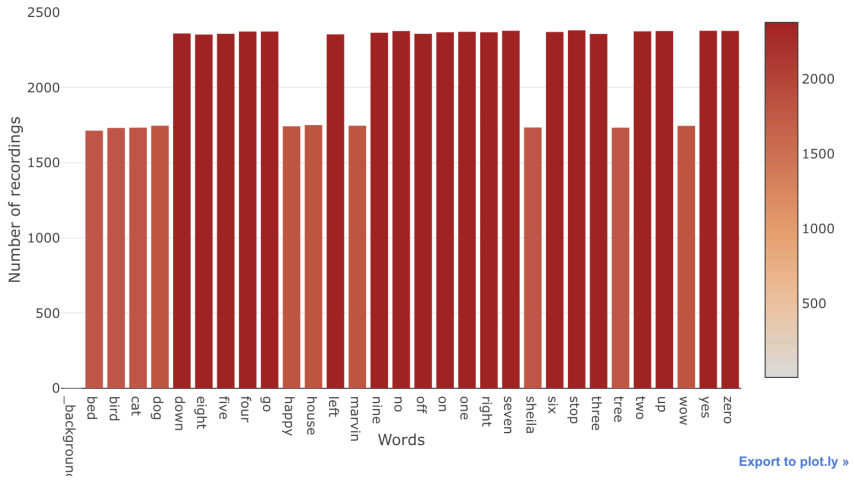
dataset contains audio from controlled environments with no external noise, just recording artifacts such as microphone buzz. Some of the limitations of this dataset are that we have to build a neural network model with Sparse Training Dataset, the Hardware limitation and trying to build a compact model.

Assumptions

The data leveraged in this project is pre-cleaned by the providing institutions. It is assumed that all utterances are correctly labeled. With the exception of Librispeech noisey dataset, all other utterances are collected from controlled environments without interfering sounds from unintended sources.

Descriptive Analysis

Figure 4. *Number of recordings of a given label.*



Methodology

Our proposed model will build on each of the approaches discussed in our literature review. Starting with data transformation, we compare several unsupervised feature extraction

methods; reviewing the output feature structure of each method informs computation trade off decisions we'll be making in the training stage. To minimize the amount of samples needed to adequately train our models, we'll implement distance metric learning on the outputs of a siamese network, providing results that maximize the distance between similar classes (which we'll call "friends" for the purpose of this paper) and different classes (or "foes").We explore models using various sample sizes to train our model, and leave out all others for validation. Lastly, to assess the generalizability of our model to unseen data, we use the validation samples in addition to real-world data collected from our community to obtain false alarm rates on models trained on the different sample sizes.

To measure our model's improvement on current industry standards (leveraging computationally expensive processes and networks requiring a large amount of training data), we'll first create a convolutional neural network to establish a baseline model with different sample sizes (200, 100, 50, 25, 10, 5, 2 samples in each category). This will inform us not only of the accuracy score to be beat, but also of where the training sample size begins to impact performance, indicating where few shot learning methods will be beneficial. With such sample size identified, we will introduce a Siamese neural network, a class of CNN, to counteract the performance deficit identified with few shot samples. Finally, our SID and KWS models can be combined to provide a speaker-aware KWS system that performs with maximum precision and less false alarms per hour.

Feature Engineering

The nature of sound signal data is complex, containing not only signals, but also information about the speaker, language, and noise. This complexity requires extensive

segmentation and feature extraction to obtain quantifiable representations to feed into machine learning models. Various methods to accomplish this have been established in audio recognition and yield different results. Because we do not know which method will provide the best response for our use case, we'll explore several options in our baseline models.

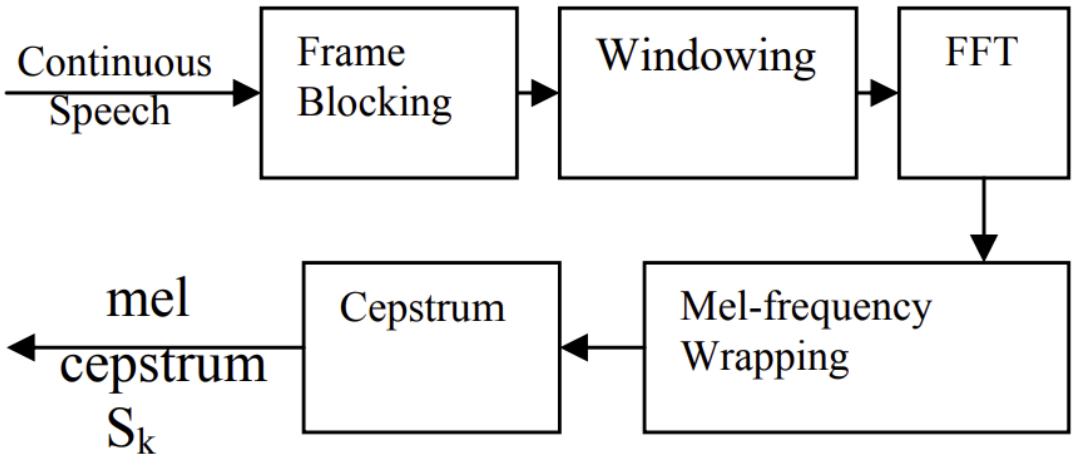
FairSeq

Similar to work by Schneider, Baevski, Collobert, and Auli, we will explore an unsupervised feature extraction method, FairSeq, before strengthening our models with supervised training. FairSeq uses a Wav2vec network to extract features in a two step process. The first step is running sound wave data through an encoder that embeds the signal in latent space, which allows for similar data points to be represented by proximity to one another. The second step involves feeding the information to the transformer.

Mel Frequency Ceptral Coefficients

Like Wav2vec, Mel Frequency Ceptral Coefficients (MFCC) is another common method of signal processing, with the goal being to transform a speech waveform into a parametric representation (Rashidul Hasan, Mustafa Jamil, Golam Rabbani, and Saifur Rahman). Audio data can be truncated into 5-100 millisecond sound bytes and fed into an MFCC network and transformed into a series of feature vectors. Sound bytes of this size are considered quasi-stationary and exhibit little change in sound wave. To create the feature vectors, sound is transformed to the Mel Frequency Scale which is composed of two primary features: linear frequency spacing for sounds below 1000 Hz and logarithmic spacing for sounds over that threshold.

Figure 5. *Process pipeline of MFCC feature extraction.*



PCA

Both MFCC and FairSeq feature creation methods result in high dimensional outputs. To counterbalance the size of these outputs and reduce stress on computational constraints, we explored model performance of these techniques in combination with principal component analysis, or PCA. PCA allows us to prune high dimensional outputs down by removing insignificant features, leaving the most necessary and heavily utilized features behind for further processing.

Modeling Framework

Architecture

For consistency with Maxim Integrated's AI architecture, executable files will organize our train and validation processes into 1) Retrieving, extracting and augmenting data sources, 2) Initializing PyTorch models leveraging Maxim's custom PyTorch Library, ai8x, 3) Model Training, and 4.) measuring computational expenses and validating performance. Following

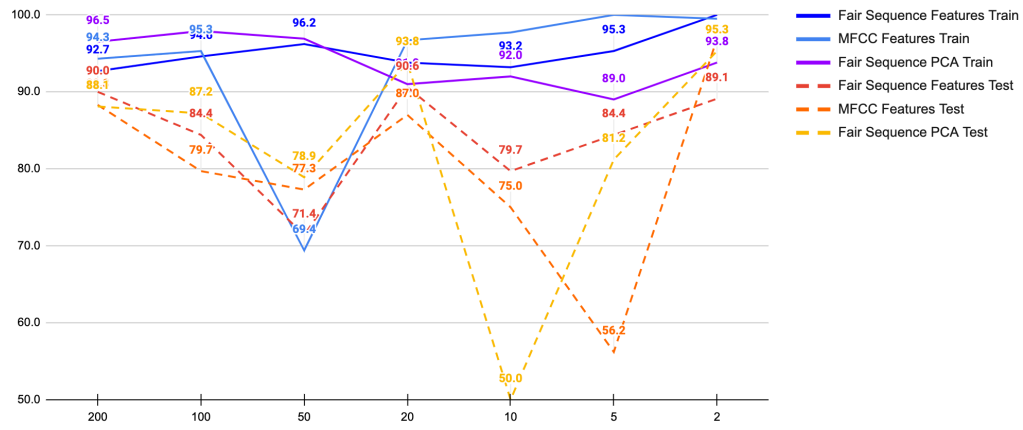
Maxim's architecture will ease company adoption of the models developed for this project and ensure smooth implementation.

The nature of the pytorch, neural networks and sheer size of training datasets requires that model training takes place over hardware equipped with Ubuntu Linux 20.04. Hardware acceleration is highly recommended. For consistent computing without local hardware limitations, data will be stored and processed using a secure remote computer provided by Maxim.

Baseline Models

The CNN baseline performance was established using three iterations of 4-layer CNN, varying feature extraction methods. This provided us with insights on the performance of models using FairSeq, MFCC, and Fairseq with PCA. For each iteration, we used a batch size of 64, a learning rate of .01, and trained for 20 epochs.

Figure 6. *Baseline Model Performance Comparison*



The first iteration of FairSeq based model has around 512 feature spaces,while the MFCC model contains around 20 features and the combined FairSeq PCA model comprises about 81

features. The 81 features in the FairSeq PCA model explain around 90% variance. For all of these models, we added dropouts on all the layers, in order for it to be generalized.

In comparing model performance on test accuracy, FairSeq with PCA performs best. From this baseline, we implement our siamese network with distance metric learning on the initial FairSeq PCA model. [insert specs on final siamese network here]. The final iteration of our siamese network had specified parameters of a learning rate of .01 for 20 epochs, where we saw our training accuracy stabilize.

Findings

Discussion

The final siamese model resulted in a test accuracy score of [XX], an X improvement on our best baseline performance of xx% on sample sizes of 10. The siamese approach to sample sizes at 5 also proved beneficial, resulting in an accuracy score of XX.

Figure 7. *Siamese Network Class Similarity Labels before Distance Metric Learning.*

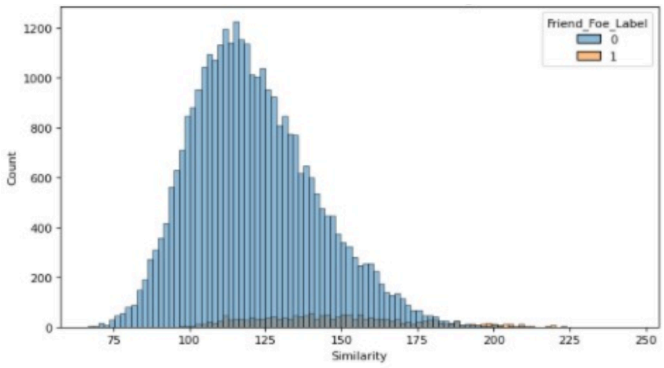
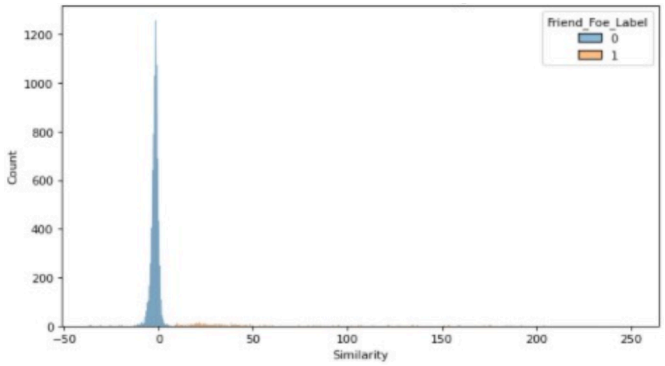


Figure 8. *Siamese Network Class Similarity Labels after Distance Metric Learning.*



Conclusion

The accuracy improvement of combined methodologies of FairSeq, PCA and Siamese over traditional models prove worth deeper exploration and implementation in the audio recognition space. Combining Distance Metric Learning techniques with shallow neural networks improves the ability of neural networks to differentiate classes of the same type from classes of different types, as indicated by the greater difference in similarity metrics. Resource heavy networks, like the more traditional CNN's used in KWS models, begin to fail when training samples are reduced below 20. While this is expected in neural networks, as they favor large amounts of training data, it's impractical and burdensome to users to provide over 20 samples of a keyword. By leveraging a siamese keyword spotting model, not only is the burden on users reduced without sacrificing model performance, but the added inconveniences and risks of cloud based competitors is also avoided.

Recommendations

Future work is recommended on integrating KWS models for a speaker aware few shot keyword spotting model. Expanding model training and tuning on new sources of data, such as

real-world collections would be a reasonable next step. Joining KWS models with additional audio recognition tools such as Speaker Identification and Noise Suppression would further improve the performance of audio recognition devices.

References

Bromley, Jane, James Bentz, Léon Bottou, Isabelle Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. “Signature Verification Using a 'Siamese' Time Delay Neural ...” Accessed November 3, 2021. https://proceedings.neurips.cc/paper/1993/file/288cc0ff022877bd3df94bc9360b9c5d-Paper.pdf.

Felzenszwalb, Pedro, Ross Girshick, David McAllester, and Deva Ramanan. “Object Detection with Discriminatively Trained Part Based Models.” IEEE Xplore . Accessed November 3, 2021. https://ieeexplore.ieee.org/document/5255236/.

Furui, Sadaoki. "Speaker recognition in smart environments." In Human-Centric Interfaces for Ambient Intelligence, pp. 163-184. Academic Press, 2010.

Gales, Mark, and Steve Young. “The Application of Hidden Markov Models in Speech Recognition.” Foundations and Trends in Signal Processing. Accessed November 3, 2021. http://www.nowpublishers.com/article/DownloadSummary/SIG-004.

Gartner_Inc. “Forecast: Internet of Things - Endpoints and Associated Services, Worldwide, 2017.” Gartner. Accessed November 3, 2021. https://www.gartner.com/en/documents/3840665/forecast-internet-of-things-endpoints-and-associated-ser.

Geron, Aurelien. “Hands-on Machine Learning with Scikit-Learn and Tensorflow.” Accessed November 3, 2021. https://upload.houchangtech.com/pdf/Hands-on_Machine_Learning.pdf.

Gu, Jiuxiang, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, et al. “Recent Advances in Convolutional Neural Networks.” Pattern Recognition. Pergamon, October 11, 2017. https://www.sciencedirect.com/science/article/abs/pii/S0031320317304120.

Han, Song, Huizi Mao, and William J. Dally. “Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding.” arXiv.org, January 19, 2016. https://arxiv.org/abs/1510.00149v4.

Hasan, Md & Jamil, Mustafa & Rabbani, Golam & Rahman, Md. Saifur. (2004). Speaker Identification Using Mel Frequency Cepstral Coefficients. Proceedings of the 3rd International Conference on Electrical and Computer Engineering (ICECE 2004).

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition.” arXiv.org, December 10, 2015. https://arxiv.org/abs/1512.03385.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. “Distilling the Knowledge in a Neural Network.” arXiv.org, March 9, 2015. https://arxiv.org/abs/1503.02531.

Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.” arXiv.org, April 17, 2017. https://arxiv.org/abs/1704.04861.

Iandola, Forrest N., Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. “Squeezenet: Alexnet-Level Accuracy with 50x Fewer Parameters and <0.5MB Model Size.” arXiv.org, April 6, 2016. https://arxiv.org/abs/1602.07360v3.

Koch, Gregory, Richard Zemel, and Ruslan Salakhutdinov. “Siamese Neural Networks for One-Shot Image Recognition.” Department of Computer Science, University of Toronto. Accessed November 3, 2021. https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf.

Merenda, Massimo, Carlo Porcaro, and Demetrio Iero. “Edge Machine Learning for AI-Enabled IOT Devices: A Review.” MDPI. Multidisciplinary Digital Publishing Institute, April 29, 2020. https://www.mdpi.com/1424-8220/20/9/2533.

Plastiras, George, Maria Terzi, Christos Kyrkou, and Theocharis Theocharidcs. "Edge intelligence: Challenges and opportunities of near-sensor machine learning applications." In 2018 IEEE 29th international conference on application-specific systems, architectures and processors (asap), pp. 1-7. IEEE, 2018.

Ravi, S, Larochelle, H. (n.d.). Optimization as a model for few-shot learning - openreview. Retrieved February 23, 2022, from https://openreview.net/pdf?id=rJY0-Kcll

Ryan, Mark D. "Cloud computing privacy concerns on our doorstep." *Communications of the ACM* 54, no. 1 (2011): 36-38.

Schneider, Steffen, Alexei Baevski, Ronan Collobert, and Michael Auli. (2020, June 10). Deep learning with Bert on azure ML for text classification. TECHCOMMUNITY.MICROSOFT.COM. Retrieved February 23, 2022, from https://techcommunity.microsoft.com/t5/ai-customer-engineering-team/deep-learning-with-bert-on-azure-ml-for-text-classification/ba-p/1149262

Tanui, Meshack. "Insecurity in the Internet of Things-Amazon Alexa." (2020)

Weinberger, Killian, and Lawrence Saul. (2009). Distance metric learning for large margin nearest neighbor Classification. Journal of Machine Learning Research. Retrieved February 24, 2022, from https://jmlr.csail.mit.edu/papers/volume10/weinberger09a/weinberger09a.pdf

Appendix A: Relevant Definitions

KWS - Keyword Spotting. Keyword spotting is a problem that was originally defined in the context of speech processing. In speech processing, keyword spotting deals with the identification of keywords in utterances. Keyword spotting is also defined as a separate, but related, problem in the context of document image processing

SID - Speaker Identification. Algorithm based voice identification

MI - Maxim Integrated

CNN - Convolutional Neural Networks.

AI - Artificial intelligence

Masking - the effect on certain sounds that can no longer be interpreted when combined with an additional sound.

Time Warping - A random point will be selected and warping to either left or right with a distance W which is chosen from a uniform distribution from 0 to the time warp parameter W along that line.

Appendix B: Implementation Plan

Table 2. *An Estimated Timetable of tasks to be completed this quarter, with additional high priority benchmarks through the end of the project.*

Topic	Task	Deadline
Descriptive Analysis	Understand the demographics of the data, Audio analysis of different: ● Gender	7th Nov 2021

	• Age groups • Ethnicity Understand the biases in the data due to demographics.	
Data Preparation	Clean the data by removing unwanted audio parts	14th Nov 2021
Data Extraction	Extracting and augmenting the features	14th Nov 2021
Modelling Framework	Understanding the Embedding Approach	21st Nov 2021
Model Preparation	Preparing the data for Model	24th Nov 2021
Model Training	Trying for different Iterations of the model	28th Nov 2021
Initial Findings	Benchmarking different models	6th Dec 2021
Model Training	Train the model with multiple datasets	20th Dec 2021
Model Validation	Model will be validated against the test set	30th Dec 2021
Model Assessment	Model assessment against analysis goals to be achieved	15th Jan 2022
Final Report Draft	Final draft to be completed and submitted for assessment	25th Jan 2022

Final Report	Final report to be completed and submitted for assessment	10th Feb 2022