SMU Data Science Review

Volume 5 | Number 2 Article 11

2021

Real-Time Voice Biometric Speaker Verification

Inderbir Dhillon
Southern Methodist University, idhillon@smu.edu

Jason Rupp
Southern Methodist University, jrupp@smu.edu

Aniketh Vankina
Southern Methodist University, svankina@smu.edu

Robert Slater
Southern Methodist University, rslater@smu.edu

Recommended Citation

Dhillon, Inderbir; Rupp, Jason; Vankina, Aniketh; and Slater, Robert (2021) "Real-Time Voice Biometric Speaker Verification," *SMU Data Science Review*: Vol. 5 : No. 2 , Article 11.
Available at: https://scholar.smu.edu/datasciencereview/vol5/iss2/11

Real-Time Voice Biometric Speaker Verification

Inderbir Dhillon[1], Jason Rupp[1], Aniketh Vankina, Dr. Robert Slater

[1] Master of Science in Data Science, Southern Methodist University,
Dallas, TX 75275 USA

Abstract. Automated speaker verification has been an area of increased research in the last few years, with a special interest in metric learning approaches that compute distances between speaker voiceprints. In this paper, three metric learning systems are built and compared in a one-shot speaker verification task using contrastive max-margin loss, triplet loss, and quadruplet loss. For all the models, spectrograms are created from speaker audio. Convolutional Neural Network embedding layers are trained to produce compact voiceprints that allow users to be distinguished using distance calculations. Performances of the three models were similar, but the model with the best EER used triplet loss in this experiment.

1 Introduction

Speaker verification is needed whenever someone wants to access an account remotely. Many institutions, especially financial institutions, allow users to access their accounts via telephone. This method is usually a tedious and time-consuming (therefore costly) part of the interaction. Recent advances in automated speaker recognition have led to more interest in automating speaker verification systems. This paper constructs a speaker verification model using different loss functions: contrastive pairwise loss, triplet loss, and quadruplet loss.

Automated speaker verification entails comparing the utterance of an unknown speaker to a voiceprint of a single speaker. This is a one-to-one comparison where the question is simple: are these two audio segments from the same speaker. This differs from speaker identification because identification entails a one-to-many comparison between an unknown speaker's utterance and multiple voiceprints. Speaker verification is a more straightforward task due to the relative simplicity, so there is a real promise that an automated verification system can be implemented given the previous success of blacklist speaker identification systems [1].

Successful production systems have traditionally utilized post-hoc processes whereby known fraud voices are used to create voice models of known fraudsters. High-risk calls can then be screened to identify fraud callers, and the institution can put a hold on any transactions related to the call. In this type of production system, screening is typically not done in real-time, and knowledge-based authentication is still needed.

Additionally, many fraudulent calls are required to continually update the fraudster profiles because the quality has a serious impact on system performance. Designing an automated system that could effectively screen calls without needing a large knowledge bank of fraudulent voices could alleviate several of these problems.

The challenges to implementing speaker verification in the past have been a lack of training data and a lack of attention to one-shot learning. Theoretical gains have been made in speaker recognition tasks such as the National Institute of Standards and Technology (NIST) Speaker Recognition Evaluation providing the underpinning for a production automated speaker verification platform. Still, additional work needs to be done to produce a system that is performant enough for real applications.

This paper builds on more recent work, which has produced highly performant speaker verification systems using large datasets such as the Vox Celeb dataset and sophisticated metric learning loss functions. Of particular interest in this paper is comparing the performance of three different loss functions: contrastive pairwise loss, triplet loss, and quadruplet loss.

Additionally, real-world systems don't have the advantage of requiring collections of verification voiceprints. A system that requires multiple verification voiceprints is more cumbersome and adds marginal utility. This study aims to use voice biometrics to build a real-time text-independent speaker verification system that can verify identity-based on a single past voiceprint.

The remainder of this paper is organized as follows: a review is done of past techniques in speaker verification in Section 2; a breakdown of the contrastive pairwise, triplet, and quadruplet models built in this model is done in Section 3; the results of these models are presented in Section 4; Section 5 consists of a discussion of the model as well as the ethical concerns of this research; the work is concluded in Section 6. The code used in this paper can be found at the following link: https://github.com/IndyD/Speaker-Verification-Capstone.

2 Literature Review

The following segments will give an overview of the theory and results of previous work in speaker verification.

2.1 Theory

2.1.1 Pre-Processing

One of the keys to finding a meaningful representation of a voice in the data pre-processing step. To convert the audio segments for speech recognition, Fourier transforms used to covert the data to the frequency space, which is much more helpful in evaluating the recurring characteristics in an audio segment than the raw signal. From a speech signal, the power of the frequency needs to be assessed and can be visualized through spectrograms.

The Fast Fourier Transform (FFT) is one form of Fourier transform that can be used to determine the power of a sampled frame at different frequencies. In essence, the Fourier transform is a tool used to reconstruct periodical waveforms using series harmonics and their multiples [2]. When mentioning Fourier transformation in general, Discrete Fourier Transformation (DFT) is another contender. The reason that FFT is used over DFT is that it is more computationally efficient [2].

2.1.2 Feature Extraction

Once the data has been pre-processed, the feature extraction process starts, and the frequencies are evaluated using FFT and spectrograms. Feature extraction is the next step in implementing a successful speech recognition system. Feature extraction is one of the most important aspects of speaker identification. A well-chosen feature representation can make discrimination of speakers much easier.

Feature Extraction Algorithms by Alim and Rashid (2017) discuss the importance of these techniques, including but not limited to Mel-Frequency Cepstral Coefficients, Linear Prediction Cepstral Coefficients, Line Spectral Frequencies, Discrete Wavelet Transform, and Perceptual Linear Prediction. The main purpose of feature extraction is to illustrate a speech signal by getting a predetermined number of signal components [1]. This is done by "…changing the speech waveform to a form of parametric representation at a relatively lesser data for subsequent processing and analysis" [1].

MFCC (Mel-Frequency Cepstrum Coefficients) is a technique often used in feature extraction before neural network embeddings. When computing MFCC, the spacing of Mel-filter banks and choice for the number of Mels becomes a key concern. The filter banks are used to capture the energy of a voice into different discrete bins.

MFCCs denote low-frequency regions better than a high-frequency region [1]. Hence, it has better compute power for formants in the low-frequency range and can describe the vocal tract resonance [1]. In addition, MFCC was previously viewed as the technique of choice for general speaker identification applications because it has a reduced vulnerability to noise disturbance [1, 3].

MFCC was the state-of-the-art feature extraction technique for many years but eventually got surpassed by Neural Network embeddings partially because MFCCs have many parameters that need tuning [6]. Neural Network embeddings can use the Mel-filter bank data and find the best transformation for feature extraction.

2.1.3 Speaker Identification with Statistical Techniques

Once feature extraction has been done on the audio signal, a model is needed to identify speakers. Early success in the field of speech verification was achieved using Gaussian Mixture Models (GMM). In this approach, the audio signals can be considered a Gaussian mixture of two factors: the audio features associated with the particular speaker and the audio features related to the channel [8]. Joint Factor Analysis (JFA) was used to model the expected variability within the channel and the audio features within a speaker's voice as separate subspaces. The JFA models the speech features as a linear combination of the channel/session subspace, the speaker subspace, and a Universal Baseline Model (UBM) to isolate the unique speaker's variability [8].

The UBM is generated from a diverse set of baseline voices, which should represent the population being screened, so the amount of variability expected in a subset of generic speakers is captured. An individual speaker's subspace is distinct from the UBM (the portion of the variability in the speaker's voice that is not captured in the UBM) and can be used to identify the unique speaker.

The JFA approach was superseded by a single factor extraction technique introduced by Dehak et al. in 2009 called i-vector extraction, which remained state-of-the-art until the recent challenge by neural network approaches [8]. Analogous to JFA, the i-vector extraction consists of a GMM-UBM. However, all the remaining variability is modeled within a single remaining term compressed to a lower dimension. The low-dimensional representation of this remaining variability term is known as the i-vector. Since the UBM had been removed, the i-vector should only contain the features identifying an individual's voice and the channel features [8].

After creating i-vectors, another step is needed to maximize the difference between channel variability and speaker variability in the i-vector. Dehak et al. initially used Linear Discriminant Analysis (LDA) that projects the i-vector to the space with the largest separation [9].

Rather than doing LDA, a simple cosine distance scoring can be calculated between the channel-compensated i-vectors instead of relying on a classification model. This cosine distance approach allows very efficient audio screening compared to the JFA approach, which would be very useful in a real-time screening application [9].

Much work has gone into maximizing the difference between channel variability and speaker variability in i-vector speaker verification. Yao et al. (2018) explored using different channel compensation techniques using the RSR2015 speaker evaluation database. They found marginal improvement over LDA when using Gaussian probabilistic LDA (EER 5.14 and 4.79, respectively). However, there was a more significant improvement when using neural networks. Using a speaker classifier network (SCN), discriminative deep metric learning (DDML), and discriminatively learned network (DLN) yielded EERs of 4.26, 3.89, and 3.44, respectively [10].

2.1.4 Deep Neural Networks

In recent research, neural networks have significantly outperformed traditional channel-compensation techniques for classification, and neural networks are increasingly being used to extract the features as well. Using neural networks in speaker recognition has grown in popularity, especially now that there is ample access to computing power.

Deep Neural Networks (DNN) can be employed for speaker verification, which is are neural networks with several hidden layers. These networks are comprised of an input layer, which can be composed of several features. The features will have weights applied at each layer, in addition to an activation function. Lastly, the neural network will have an output layer, which is exactly as the name implies.

As described in Richardson et al. (2015), two general ways to apply a deep neural network to identify speech are an indirect method and a direct method. In the direct method of DNN speaker verification, a trained DNN is used to determine the speaker. The indirect method differs in that a second DNN is employed. The first DNN is used to extract features from the input data. This extraction neural network does not have to be trained specifically for this task; it could have another purpose. Once the features have been extracted, they can be fed into a secondary classifier, a second DNN trained specifically for speaker verification.

One feature extraction method by a DNN is accomplished by one of the hidden layers acting as a speaker representation. This technique employed by Richardson et al. (2015) and Kydyrbekova et al. (2020) involves a bottleneck layer. This hidden layer of the DNN has fewer nodes than the surrounding hidden layers. What this does is force the features to compress. This is not all dissimilar to principal component analysis and linear discriminate analysis. After the features pass through the bottleneck layer, further transformations create new features used for speaker verification.

Some have taken a hybrid approach with speaker recognition and verification. After processing the initial parts of speech and extracting the converting the speech into an MFCC feature set [12], these are fed into the DNN.

Pre-processing of the data is not required when using DNNs for speaker verification [13]. Unprocessed training datasets can be fed directly into the DNN, and the model controls for channel-specific normalization while still outperforming traditional statistical modeling techniques.

2.1.4 Convolutional Neural Networks

A recent technique that has gained wide adoption is to fit a Convolutional Neural Network to spectrograms of short audio segments of speaker utterances to perform end-to-end feature extraction and speaker recognition. This can be seen as stretching the theoretical foundations of CNNs since the image processing techniques were not intended to serve dimensionality reduction on spectrograms. Still, Nagrani et al. (2020) were able to achieve state-of-the-art performance on the VoxCeleb dataset using a 2-dimensional CNN based on the ResNet architecture [14].

ResNets are a breakthrough in image classification techniques that allow the training of much deeper networks without degradation in performance. They do this by adding "shortcut connections" that skip one or more layers, helping overcome some of the issues that traditional deep neural networks encounter [15]. The shortcut connections are "identity mappings" rather than parameters and do not add computational complexity. Due to these "shortcut connections," ResNets can build larger, deeper neural networks that are similarly performant to other state-of-the-art image techniques like the Visual Geometry Group (VGG) and achieved state-of-the-art recognition on ImageNet in 2015 [15].

VGG Neural Networks are a type of ConvNet model that uses a small convolutional filter (3 by 3) and a Relu activation function in all the layers, which allows for deeper ConvNets than were previously possible [16]. VGGs achieved state-of-the-art on the ImageNet 2014 challenge and have remained one of the most popular computer vision techniques [16].

Researchers have studied many techniques for automated speaker detection. Still, less attention has been paid to the case where a single verification audio sample is used for speaker verification, known in many applications as "one-shot learning." One-shot speaker verification can be particularly challenging because channel-specific effects aren't averaged out across multiple samples. Velez et al. (2018) tested several architectures for one-shot speaker identification using Siamese Convolutional Neural Networks that succeeded in applying service robots despite requiring a one-to-many comparison [17].

Siamese Neural Networks are an architecture where a pair of neural networks with similar (or the same) weights are constructed [18]. Pairs of audio samples from matching and non-matching speakers are fed to the Siamese Neural Network. A loss function such as contrastive loss or triplet loss is used to minimize the distance between matching pairs and maximize the space between non-matching pairs. Siamese Neural Network architectures are particularly well suited to one-shot learning because they are inherently designed to compare pairs [18].

2.1.5 Contrastive Loss Functions

As previously mentioned, a metric is needed that calculates the distance between voices rather than treat this as a classification problem. To that end, a loss function is needed that compares multiple inputs to compute the distance between them. An intuitive loss function for this application is max-margin contrastive loss.

Contrastive triplet loss is more complicated but has been used in state-of-the-art metric learning systems. Contrastive quadruplet loss is an extension of triplet loss with another input and will be explored in this paper.

Max Margin Contrastive Pairwise Loss

Max margin contrastive pairwise loss is a loss function that takes two data inputs and a label. Here is the formula:

$$\boldsymbol{loss(d, Y) = \frac{1}{2} * Y * d^2 + (1 - Y) * \frac{1}{2} * max(0, m - d)^2}$$

Here *Y* is the label (0 is the inputs are from different speakers, one otherwise). The distance calculated between the two inputs is d, and there is a margin parameter *m* [19].

The label term is used as an indicator. If the label is 1, the loss is the squared distance between the two inputs. Intuitively, distances between audio samples from the same speaker are considered a loss since they should be very close together. When the label is 0, the inputs should be forced at least m units away from each other. If they are closer than m, the loss is the square of the extent to which the inputs violate the distance requirement. In this way, audio segments for non-matching speakers are punished for being too close together.

Contrastive Triplet Loss

Contrastive triplet loss has a slightly different setup in that no label is provided. Instead, every data record contains three inputs. The first input is the baseline (called the anchor). The following two inputs are a similar example (i.e., another audio from the same speaker) or a dissimilar example (i.e., audio from a different speaker). Here is the formula for triplet loss:

$$\boldsymbol{loss(d_1, d_2) = max(d_1^2 - d_2^2 + m, 0)}$$

Here, d_1 is the distance between similar inputs and d2 is between dissimilar inputs. There is still a margin parameter *m* as well. This loss function adds loss when the dissimilar inputs are less than *m* units further apart than the similar inputs [20].

Contrastive Quadruplet Loss

The final loss function covered in this paper is contrastive quadruplet loss, which has the formula below:

$$\boldsymbol{loss(d_1, d_2, d_3) = max\left(d_1^2 - d_2^2 + m, 0\right) + max(d_1^2 - d_3^2 + m, 0)}$$

The d1 and d2 terms represent the distances between similar and dissimilar inputs, respectively, like triplet loss. Here, however, there is also a d_3 term which is the distance between the anchor and a second dissimilar inputs (which is dissimilar from input d_2) [21].

The loss function is essentially the combination of two triplet loss functions, one for separating d_1 from d_2 and another for separating d_1 from d_3. Both tasks have their margin, m_1 and m_2; they can be set to the same value in practice.

2.2 Models

2.2.1 MFCC

Research into MFCCs and other spectral feature extraction techniques generally revolves around tuning the parameters of these features, such as frame size and frequency filters. The performance of MFCCs can vary substantially based on the choice of these parameters, which can be seen as a disadvantage of using them for feature extraction.

Kopparapu and Laxminarayana wrote a 2010 article where they reviewed the results of multiple experiments to explore some of these features. The speech signal was sampled at 16 kHz and represented by 16 bits. The speech signal was then divided into frames of 32 ms and 16 ms. This resulted in 512 and 256 samples, respectively [22]. Most of the research was done on small speech samples.

The Mel filter banks were then computed for each sample speech frame on a 30-band frame from a minimum frequency of 130 Hz to a maximum of 6800 Hz [22]. A minimum of 26 coefficients are obtained, but only 12-13 coefficients are kept for speech recognition. The article does not provide the number of MFCC pulled but is implied that it was more than 26 coefficients. The choice of the Mel Filter banks computation affected the performance. These results show that the nature of MFCCs makes so that there is no single best extraction but context-dependent.

2.2.2 Statistical Models

A big advancement in automated speaker recognition was a 2004 study. Zheng et al. used a recently introduced GMM-UBM speech recognition architecture to significantly improve the existing GMM based text-independent speaker recognition model on the 2000 NIST Speaker Recognition Evaluation corpus using MFCCs for feature extraction. The GMM-UBM model outperformed a baseline model based on GMMs without a Universal Background Model, reducing relative error by 31.2%. The GMM-UBM had a test error of 29.8%, an impressive level at the time [23].

By 2006, Dehak et al. significantly outperform the GMM-UBM joint factor analysis approach with an i-vector approach. They experimented with several improvements to the architecture and found that the best results were obtained using Within-Class Covariance Normalization (WCCN) and LDA for the final classification. In the NIST 2006, Speaker Recognition Evaluation set, the i-vector with WCCN had an Equal Error Rate (EER) of 2.7%, compared to 3.8% with the JFA approach. An advantage of this approach is that a simple cosine distance scoring was calculated between the channel-compensated i-vectors, so audio screening could be done more quickly than the JFA approach, which would be very useful in a real-time screening application [9].

There is continued research into the i-vector approach. In a 2017 study, Kanrar uses a cosine-based prediction model that used data collected from a recording from a railway station in India in Hindi, Bengali, Teague, and Oriya [24]. Most of the test was from 45-second intervals, with a list of 30 people. An i-vector with 400 dimensions equaled to 39 MFCC features was created for automated speaker recognition [24]. The results achieved 80% accuracy in terms of verified speakers, which is an impressive result given the model's simplicity.

2.2.3 Neural Net Models

While some researchers have continued with i-vector research, the more recent effort has gone to CNN approaches. In a 2020 study, Nagrani et al. tested various models, including 34-layer ResNets, 50-layer ResNets, and a modification of ResNet called ThinNet that has fewer parameters. In the end, the 34-layer ThinNet had the lowest EER of 2.87%, achieving state-of-the-art on the VoxCeleb2 dataset [14]. The contrastive loss was used, but due to the difficulty of finding convergence when training solely with contrastive loss, the model was trained for identification using a softmax loss function then the classification layer was replaced with a fully connected layer and finished training with the contrastive loss [14].

Velez et al. (2018) tested Siamese CNNs with both ResNet and VGG architecture. A Voice Activity Detector was used for both networks to detect when the speaker was speaking, and a one-second audio sample was captured and turned into a spectrogram. Speaker identification was then done against a verification set chosen from the VoxCeleb dataset [25]. The speaker was identified by performing a speaker verification task. Surprisingly, a simple 7-layer VGG network outperformed a 50-layer ResNet with accuracies of 91% and 89%, respectively, in the task of speaker identification [25].

A key to this architecture is that the Siamese Network allows for one-shot learning where a speaker can be identified based on a single previous speech segment. Additionally, a new voice sample can be added for an identification task without retraining the whole network. These are the operational concerns that are often overlooked in studies. This paper creates an end-to-end real-time speaker verification system for the whitelist voice biometric use case that requires only one voiceprint per speaker.

3 Methods

3.1 Data

The data used for this paper was the Voxceleb dataset, a compilation of audio and video files obtained from interviews of celebrities posted on YouTube. This dataset includes over 6000 voice samples of these celebrities and over one million utterances encompassing over 2000 hours of audio. This dataset has been increasingly used in speech research because it contains more speakers than the NIST Speaker Recognition Evaluation dataset and is freely available. Voxceleb audio segments were obtained from YouTube videos of celebrities giving interviews, often in front of audiences. This means that there is potential for the background noise as well as interruptions by other speakers. This makes training more complicated but can lead to more robust models for real-world applications where these issues must be accounted for.

3.2 Implementation

For speaker validation, a generic speaker embedding model was built based on the VGG7 architecture in [17]. For each speech segment, a spectrogram is created from 5 seconds of speaker audio. From there, models are trained using contrastive pairs loss, triplet loss, and quadruplet loss to compare the performance of these different models. These models take various inputs, but they all have the same CNN embedding model: the first seven layers of VGG16. This method has been successful for researchers in the past [17] [25].

This CNN embedding architecture can be seen in Figure 1. There are four total convolutional layers, two dropout layers, and one fully connected layer.

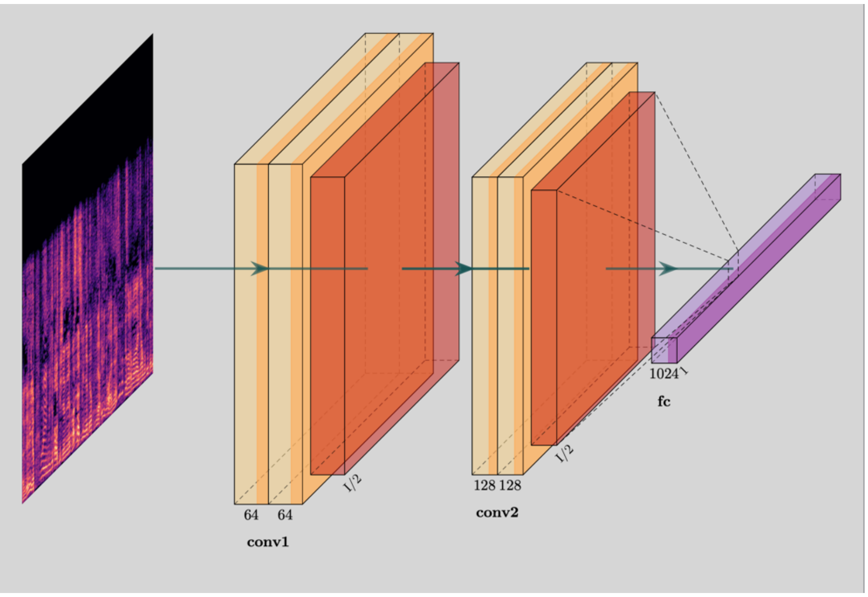

Figure 1. VGG-7 Embedding Model

The Siamese, Triplet, and Quadruplet models shown in Figures 2-4 use these embedding layers as the base and add a layer that compares the Euclidean distances between the inputs. The Siamese model takes two input spectrograms and a label, the Triplet model takes three spectrogram inputs, and the Quadruplet model takes four spectrograms as input.

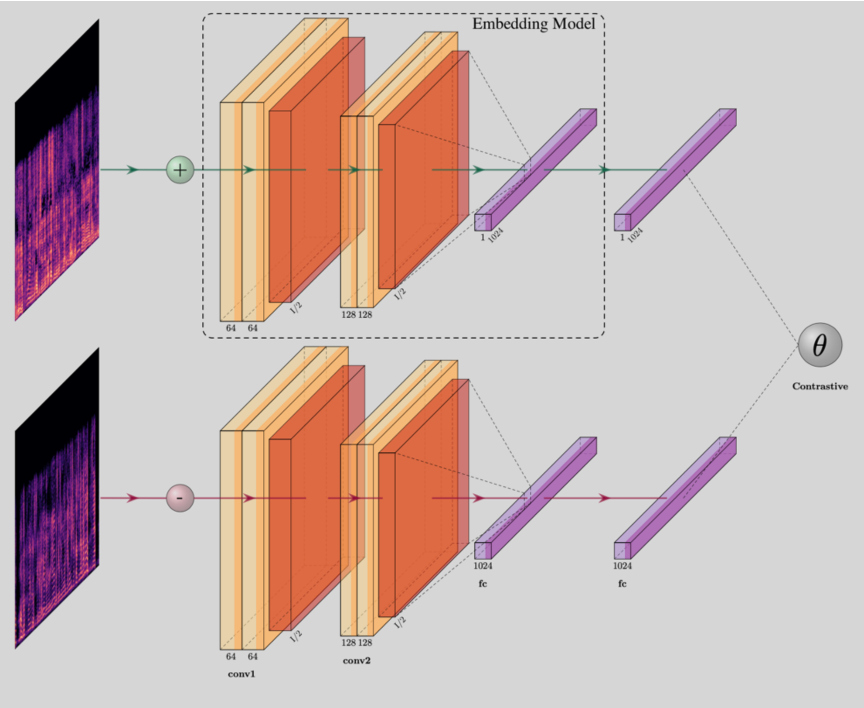

Figure 2. Pairwise Contrastive Loss Model

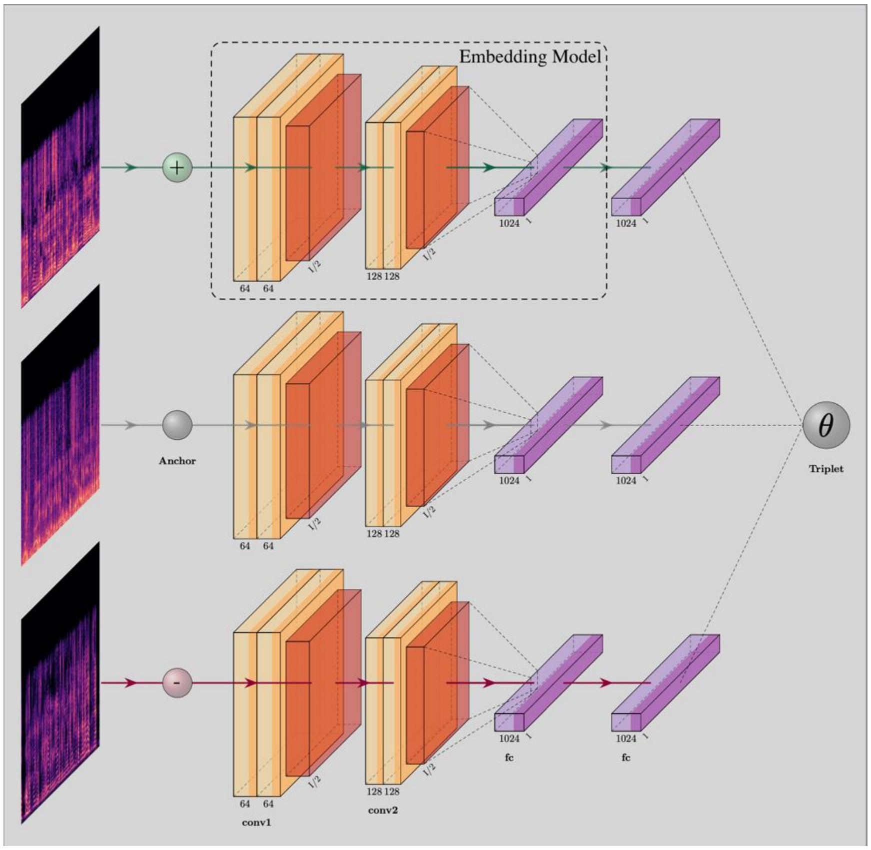

Figure 3. Triplet Loss Model

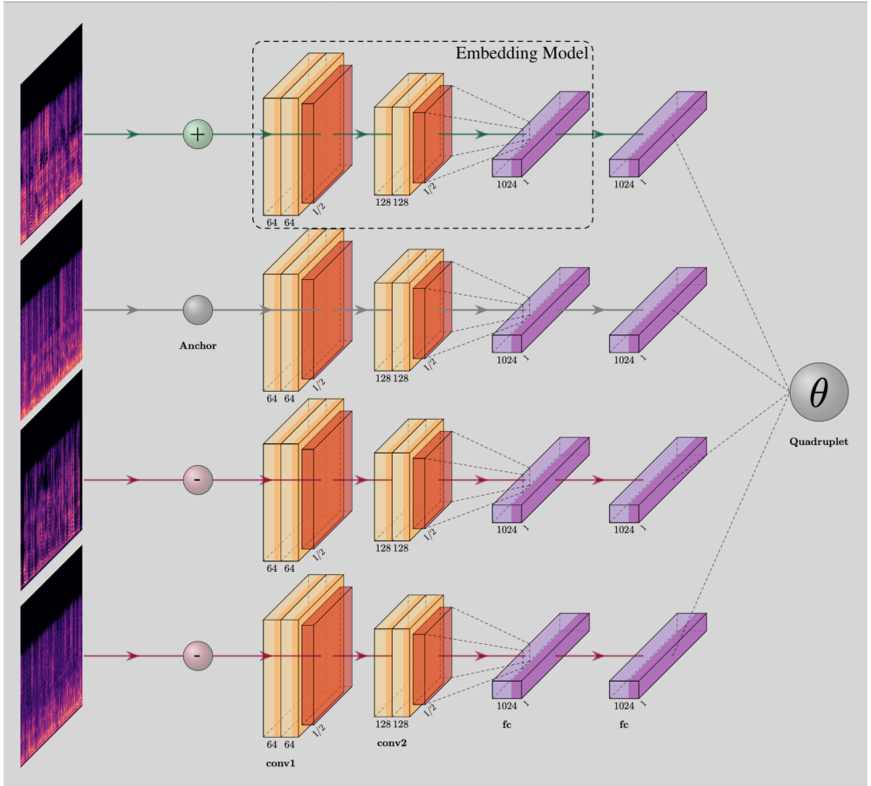

Figure 4. Quadruplet Loss Model

The output of the Siamese model trained with contrastive loss is well suited for the validation task at hand: two spectrograms are given as input, and the model scores the likelihood of matching. A final postprocessing step must be done for the Triplet and Quadruplet models to drop the final comparison layer and use the last dense layer as a "speaker model." When using the system for further identification, the Euclidean distance is taken between speaker spectrograms, and a threshold can be chosen for labeling matches. In practice, the same thing should be done with the Siamese model to control the specificity and sensitivity of the model generates.

A pre-training step was added since distance metrics such as contrastive pairs, triplet loss, and quadruplet loss cannot converge properly. In this pre-training step, the same VGG-7 embedding is used as the full models, but the last layer is a single full connected node with Sparse Cross-Entropy Loss. This becomes a multiple classification problem where every user in the corpus is a label. The embedding model gets trained to identify the speaker based on their spectrogram correctly. Once this is done, the final layer is replaced with the loss layer of interest, and further training is done to fine-tune the model.

Semi-hard mining was added as an enhancement for both triplet mining and quadruplet mining. There is some concern that these models over-optimize the easy examples and fail on complex, more interesting examples. To combat this, these models are first run with a naïve dataset to train. Once the model is trained, triplets or quadruplets that are semi-hard can be identified: anchor closer to the positive spectrogram, no negative spectrograms with 0 loss. Once a dataset of only semi-hard triplets or quadruplets is built, the model is retrained with this set.

3.3 Model Parameters

The parameters of this model fell into three main groups: spectral parameters, model parameters, and training parameters. The spectral parameters are the features used to generate the spectrograms (example in fig below) and were shared across all the models. Through experimentation, the best spectral features were found to be: 130 Mel banks, 512 FFTs, a hop length of 222, 300 max frames, and a window length of 512.

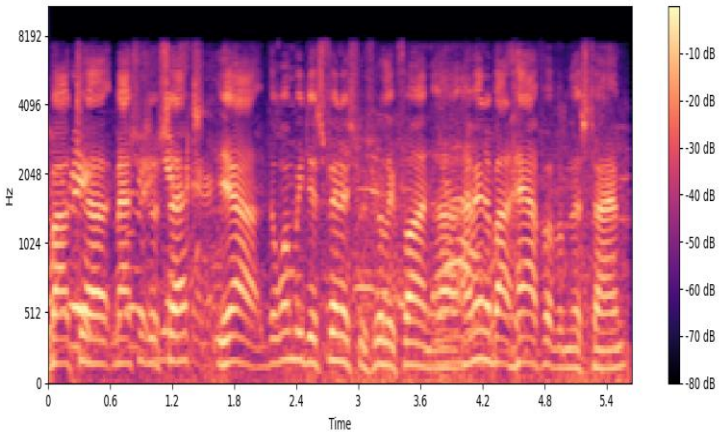

Figure 5. Sample speaker spectrogram

Once the spectrograms were created for each speaker, datasets of size 200,000 were made for each model. Ten percent of the data was held out for a test set, and a further 10% of the training set was used for validation during training. The speaker in the test set, training set, and validation set were non-overlapping to avoid overfitting.

A simple labeled set was generated for the cross-entropy training pre-training step, but each of the distance-based models required the generation of a distinct dataset. A balanced set of matching and non-matching pairs with labels was generated for the contrastive pairs model. The Triplet and Auadruplet models did not need balancing since each data point contains positive and negative examples.

A batch size of 100 was chosen during training, and training was done until there was a lack of improvement in validation loss for three epochs. Stochastic Gradient Descent was selected as the optimizer with a learning rate of 0.0001. The learning rate decayed by a factor of 0.99 over 100000 steps, and there was a momentum term of 0.9. All these values were mirrored in the pre-train phase.

4 Results

For the final evaluation of the models, a balanced test set of matching and non-matching pairs of spectrograms with labels was used for speaker validation. For all three models, the final layer of the model is removed, and only the speaker model generation layers are kept. The outputted distances can be thought of as scores. The equal error rates (EER) of false positives and false negatives can be calculated using these scores and the labels. This will serve as the primary success metric in this paper. Below is a table of the performance of various models:

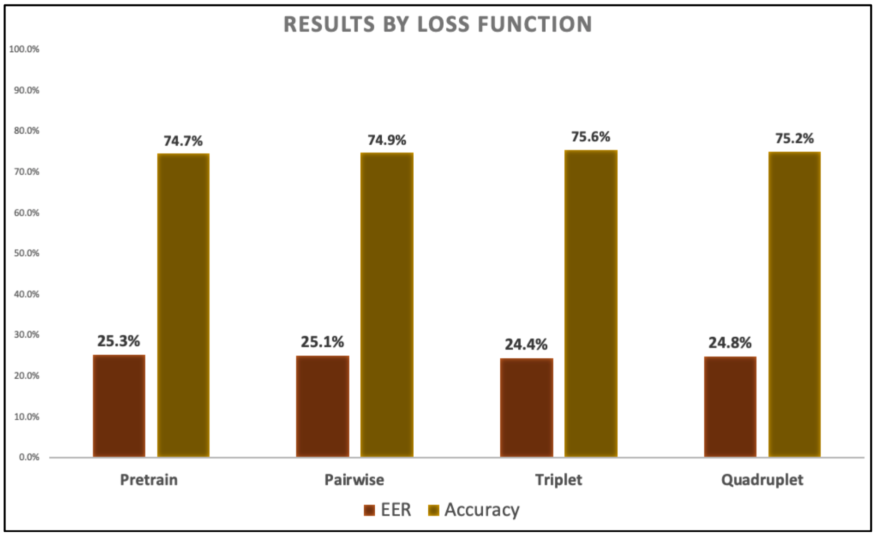

Figure 6. Model EER

Contrastive triplet loss is the best performing model in these experiments, with an EER of 24.4%. This beat out the cross-entropy classification model with produces an EER of 25.3%. The contrastive pairwise loss and quadruplet models did perform slightly better than the cross-entropy classification model with EERs of 25.1% and 24.8%, respectively.

5 Discussion

The speaker verification system presented in this paper was a successful implementation. However, the results were far from state-of-the-art. This was mainly due to a lack of tuning in the data pre-processing due to computation limitation.

The primary purpose of this research was to compare the difference between contrastive loss, triplet loss, and quadruplet loss. In general, research has shown triplet loss to be superior to contrastive pairwise loss, and quadruplet loss has certain benefits over triplet loss but has not been primarily adopted. Triplet loss did prove to be the most successful model in terms of EER.

Beyond EER, there is a tradeoff in training time and memory requirements when scaling up loss functions. Pairwise loss uses two images for each data point, whereas triplet and quadlet use 3 and 4 images, respectively. This means that triplet loss training time and memory requirements are 50% higher for triplet loss than pairwise loss and 100% higher for quadruplet loss. This limitation also supports the idea that triplet loss is a good balance between accuracy and computational efficiency.

5.1 Implementation

Real-world implementation of an automated speaker verification system requires additional components. The model presented in this paper finds distances between generic voices. Once a distance is generated, a threshold must be set to determine which voice samples are considered a match. Setting this threshold depends on the security risk of allowing the wrong speaker through. Speakers that are considered a match can be let through, while non-matching speakers would require additional verification.

Collecting audio samples require a database system that checks if the account holder has a voiceprint on file. If not, then the voiceprint is collected during the account holder's first caller. Subsequent calls can be screened against this voiceprint.

A significant benefit of using a speaker embedding model is that the last dense layer of the neural network can be thought of as the voiceprint. In this paper, the generated voiceprints are numeric vectors of length 1024. This means that speaker voices can be stored on small disks drives. Comparing voices is also computationally efficient: simply taking the Euclidean distance between voiceprints.

The proposed system has the advantage of working on top of existing fraud prevention solutions such as traditional knowledge-based authentication and blacklist screening. Any calls that are not exact matches can go through knowledge-based authentication. All high-risk calls can still be screened against a blacklist to look for possible matches to blacklisted fraudsters to screen for additional screening as well.

A concern with any automated security feature is the possibility of hostile actors. Since the models in this paper were only trained on a closed set of speakers and conditions. There is a possibility to produce sounds or that produce artificially high scores that fool the model. To some extent, this issue can be mitigated by constantly updating the training model with these examples. This would be sufficient for most applications, but further interventions can be taken, such as having the phone agent manually flag any calls with excessive noise or training as a separate model for this task.

5.2 Ethics

When dealing with the storage and usage of biometric information, consent and data security are significant concerns. In most jurisdictions, consent is required before recording a person's voice. Financial institutions and call centers usually have the infrastructure in place to handle this. Often, there is a pre-recorded message or a prompt by the phone agent informing the caller of the recording.

Given the existing infrastructure for informed consent in existing call banks, gaining consent is not a significant issue in speaker verification. A more substantial concern is around data breaches and the ethics of storing biometric information. Data breaches are a common occurrence in the modern internet landscape. When passwords as leaked, data is compromised, but the password can be changed. However, biometric voice data is fundamental to how a person speaks and is not changeable. This raises troubling questions about the idea of permanently compromised individuals.

A well-functioning voice biometrics verification system should be able to get around some of these thorny issues. First, any entity that stores biometric information of any kind should practice the highest level of data security. Even in a data breach, the advantage of speaker embedding approaches is that the data is not useful in circumventing the system. Even if hackers were to capture the voice models for all users, there is no clear way to generate the input to the verification system. Additionally, voiceprints are specific to a specific model and do not compromise a person's voice in general.

6 Conclusion

Overall, this paper successfully implemented real-time voice biometric verification comparing contrastive pairwise, triplet, and quadruplet loss. The triplet loss model was the best performing, with an EER score of 24.4%. Optimizations such as pre-training with cross-entropy loss on a classification problem and mining of semi-hard triplets let slight improvements in performance but use the last layer of the cross-entropy pre-training step as a voice model proved surprisingly effective. The result of this paper is to provide lessons about loss function performance in metric learning that can be further evaluated in future research.

References

1. Ajibola Alim, S. and Khair Alang Rashid, N., 2017. Some Commonly Used Speech Feature Extraction Algorithms. [online] https://www.intechopen.com/. Available at: https://www.intechopen.com/books/from-natural-to-artificial-intelligence-algorithms-and-applications/some-commonly-used-speech-feature-extraction-algorithms
2. Lin, H., & Ye, Y. (2019). Reviews of bearing vibration measurement using fast Fourier transform and enhanced fast Fourier transform algorithms. Advances in Mechanical Engineering, 11(1), 168781401881675–. https://doi.org/10.1177/1687814018816751
3. Chakroborty S, Roy A, Saha G. Fusion of a complementary feature set with MFCC for improved closed set text-independent speaker identification. In: IEEE International Conference on Industrial Technology, 2006. ICIT 2006. pp. 387-390
4. Chu S, Narayanan S, Kuo CC. Environmental sound recognition using MP-based features. In: IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE; 2008. pp. 1-4
5. El Choubassi MM, El Khoury HE, Alagha CEJ, Skaf JA, Al-Alaoui MA. Arabic speech recognition using recurrent neural networks. In: Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology (IEEE Cat. No.03EX795). Ieee; 2003. pp. 543-547. DOI: 10.1109/ISSPIT.2003.1341178
6. Putra, B. and Suyanto, 2011. Implementation of Secure Speaker Verification at Web Login Page Using Mel Frequency Cepstral Coefficient-Gaussian Mixture Model (MFCC-GMM). [online] www.researchgate.net. Available at: <https://www.researchgate.net/publication/229034376_Implementation_of_secure_speaker_verification_at_web_login_page_using_Mel_Frequency_Cepstral_Coefficient-Gaussian_Mixture_Model_MFCC-GMM> [Accessed 10 February 2021].
7. Ravikumar, K., Reddy, B., Rajagopal, R., & Nagaraj, H. (2008). Automatic Detection of Syllable Repetition in Read Speech for Objective Assessment of Stuttered Disfluencies. Citeseerx.ist.psu.edu. Retrieved 10 February 2021, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.307.2786&rep=rep1&type=pdf.
8. N. Dehak, et al. "Support vector machines versus fast scorring in the low-dimensional total variability space for speaker verification", in Interspeech, Brighton, UK, Sept 2009.
9. N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," in IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 4, pp. 788-798, May 2011, doi: 10.1109/TASL.2010.2064307.
10. Yao, S., Zhou, R., Zhang, P. and Yan, Y. (2018), Discriminatively learned network for i-vector based speaker recognition. Electron. Lett., 54: 1302-1304. https://doi-org.proxy.libraries.smu.edu/10.1049/el.2018.6359
11. Richardson, F., Reynolds, D., & Dehak, N. (2015). Deep neural network approaches to speaker and language recognition. *IEEE Signal Processing Letters, 22*(10), 1671-1675. doi:10.1109/lsp.2015.2420092
12. Aizat, K., Mohamed, O., Orken, M., Ainur, A., Zhumazhanov, B; Pham, D. (2020). Identification and authentication of user voice using DNN features and i-vector. Cogent Engineering, 7(1), 1751557. doi:10.1080/23311916.2020.1751557
13. Chouiekh, A., & EL Haj, E. H. (2018). ConvNets for fraud detection analysis. *Procedia Computer Science, 127*, 133-138. doi:10.1016/j.procs.2018.01.107
14. Arsha Nagrani, Joon Son Chung JS, Xie W., Zisserman A., (2020). Large-scale speaker verification in the wild. *Computer Speech & Language,* vol 60, March 2020. https://doi.org/10.1016/j.csl.2019.101027
15. K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.
16. Simonyan, Karen & Zisserman, Andrew. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 1409.1556.
17. Vélez, Rascon, and Fuentes-Pineda. (2018). One-Shot Speaker Identification for a Service Robot using a CNN-based Generic Verifier. *IEEE Robotics and Automation Letters.* e-print. arXiv:1809.04115.
18. Jadon, Shruti. (2020). An Overview of Deep Learning Architectures in Few-Shot Learning Domain. 10.13140/RG.2.2.31573.24803/1.

19. R. Hadsell, S. Chopra and Y. LeCun, "Dimensionality Reduction by Learning an Invariant Mapping," 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), 2006, pp. 1735-1742, doi: 10.1109/CVPR.2006.100.
20. Weinberger, Kilian & Blitzer, J. & Saul, L.. (2006). Distance Metric Learning for Large Margin Nearest Neighbor Classification.
21. Chen, Weihua & Chen, Xiaotang & Zhang, Jianguo & Huang, Kaiqi. (2017). Beyond Triplet Loss: A Deep Quadruplet Network for Person Re-identification. 10.1109/CVPR.2017.145.
22. Kopparapu, Sunil Kumar & Laxminarayana, M.. (2010). Choice of Mel filter bank in computing MFCC of a resampled speech. 121-124. 10.1109/ISSPA.2010.5605491.
23. Zheng, R., Zhang, S., & Xu, B. (2004). Text-independent speaker identification using GMM-UBM and frame level likelihood normalization. 2004 International Symposium on Chinese Spoken Language Processing, 289-292.
24. Kanrar, S. (2017, April 12). i Vector used in Speaker Identification by Dimension Compactness. arXiv.org. https://arxiv.org/abs/1704.03934.
25. Chung, J. S., Nagrani, A., & Zisserman, A. (2018, June 27). VoxCeleb2: Deep Speaker Recognition. https://arxiv.org/pdf/1806.05622.pdf.