Histogram Matching to Reduce Acoustic Mismatch in Automatic Speech Recognition

Cuinn Rios Fey

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science

University of Washington

2020

Committee:

Les Atlas

Mari Ostendorf

Program Authorized to Offer Degree:

Electrical and Computer Engineering

University of Washington

Abstract

Histogram Matching to Reduce Acoustic Mismatch in Automatic Speech Recognition

Cuinn Rios Fey

Chair of the Supervisory Committee:
Professor Les Atlas
Electrical and Computer Engineering

With motivation from histogram matching in image processing used to redistribute pixel probabilities in each color channel of an image, a new approach with an old technique is used for reducing acoustic mismatch between audio signals. Mel-frequency-dependent histogram matching with a silence threshold used in the log Mel-spectrogram domain is implemented before the decoding step in an automatic speech recognition system. The technique is shown to be effective within a system built to recognize low-resource, noisy, compressed, and distorted air traffic control communications. The algorithm has been shown to be robust to high acoustic variance and capable of reducing acoustic mismatch between training, validation, and test data. Additionally, it can decrease the word error rate with a statistically significant chance of confidence improvement. After tuning the algorithm's silence threshold on the validation dataset, we were able to lower the word error rate when decoding on the test dataset from 50.4% to 46.8% which is significant with $p < 0.001$.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to extend my gratitude to Steve Dame, Jasper Corleis, and Boeing for providing the data and funding for this thesis.

Dedication

This research is dedicated to my family, friends, and professors. Thank you to my family for supporting me throughout my education and cooling my restlessness with encouragement and reminders to be patient. Thank you to my friends for being open and thrilling minds to constructively interfere in my wonder and ambitions. Thank you to Kevin Everson for being a reliable and personable coworker. Thank you to Mari Ostendorf for encouraging me to pursue even higher education when I thought I had already done that. Thank you to Les Atlas for being such a profound example that our horizons are not compressed by our professions but rather expanded and for telling me a story that makes me say “no way” every time we meet.

Chapter 1
INTRODUCTION

Every voice is unique due to a plethora of biological factors. Every recording of every voice is also unique depending on factors such as room size, recording device, the presence of noise, added effects – the list goes on. This can make it difficult to compare speech in an effective way as our only way of doing so quantitatively is through a recording device. Often the differences between two voices or two different recording styles can obscure important information contained in the speech itself. The field of Automatic Speech Recognition (ASR) attempts to extract the information present in speech despite the nuances between different speech recordings; however, there is still progress to be made in improving the performance of ASR systems. In this study, it has been found that Mel-frequency-dependent histogram matching with a silence threshold used before the decoding pipeline of an ASR system can reduce this acoustic mismatch and improve the performance of the ASR system.

1.1 Background

Most ASR systems rely on either the log Mel-spectrum or Mel-Frequency Cepstral Coefficients (MFCCs) for representing the speech signal when training and decoding a neural network. An ASR neural net acoustic model is built by training the neural net weights to learn the association between the spectral representation and the phonetic information, which can be used to create text transcriptions. Unfortunately, for some tasks, engineers often lack enough accurately labelled data to train an ASR system effectively. This is known as *low-resource* ASR - referring to the low quantity of data available. Low-resource ASR introduces a host of strategies to best leverage what little data is available. One such strategy is to train the acoustic model using another rich-resource dataset that has similar properties to the low-resource dataset. The low-resource dataset is then

used to fine-tune the system [1]. This strategy allows for some elbow room in developing the ASR acoustic model; however, there may be an acoustic mismatch between the rich-resource data and the low-resource data. This acoustic mismatch can reduce the system's performance in the decoding step. This research's process of reducing the acoustic mismatch between two datasets begins with motivation from an image processing technique known as histogram matching [4][6].

1.2 Motivation: Image Processing

Histogram matching is used in image processing to redistribute a histogram distribution of binned pixel values in a source image to match the histogram distribution of binned pixel values in a reference image. By finding the linear interpolant between the cumulative distribution functions (CDFs) of the source and reference image histograms, the source's histogram bins can be redistributed. Afterwards, the redistributed source's histogram bins can be used as a key to remap pixel values, so the source and reference images are more similar. As you can see from Figure 1, the color characteristics of the cat and cup images more closely match one another after histogram matching. In probabilistic terms, histogram matching changes the probability of each source pixel value occurrence to match a reference image's pixel value occurrence probabilities.

Figure 1: Histogram matching two images. The source (cat) is matched to the reference (cup).
https://scikit-image.org/docs/dev/auto_examples/color_exposure/plot_histogram_matching.html

Looking at the histograms for each color channel shows that the source's histogram is redistributed based on the reference's CDF for each color channel.

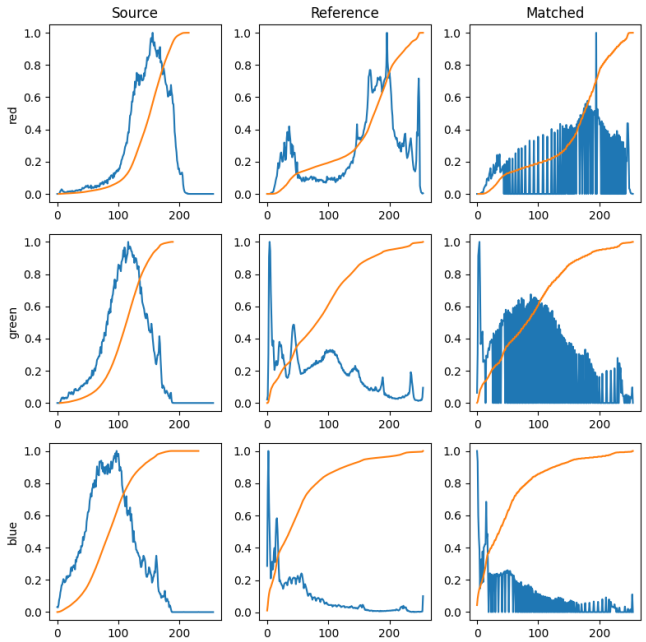

Figure 2: Histograms (blue plots) and CDFs (orange plots) of the source (cat), reference (cup), and matched (cat colored like cup).

https://scikit-image.org/docs/dev/auto_examples/color_exposure/plot_histogram_matching.html

1.3 General Overview

This section will present the technique for reducing the acoustic mismatch between two datasets prior to the decoding step in ASR. MFCCs are calculated by first calculating the spectrogram of a given speech sample, then redistributing the frequency axis to a logarithmic (Mel) scale. Next, the log magnitude is taken, and, finally, the discrete cosine transform (DCT) is applied [11], yielding the MFCCs. Figure 3 presents this process.

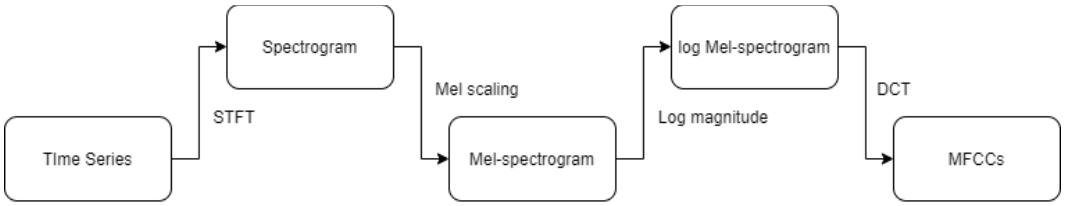

Figure 3: Process of Creating MFCCs.

If the MFCC creation process is halted at the last step before the DCT is applied, then what has been created is the log Mel-spectrogram. Reducing the acoustic mismatch between two speech recordings arises from the combination of histogram matching and the log Mel-spectrogram.

i. Naïve approach

The log Mel-spectrogram is interpreted as a single-color-channel image-like representation of a speech recording. The log Mel-spectrograms of the test and training data are treated as the source and reference images, respectively, to be histogram matched. Let it be noted that in this case the pixel values from the images are *not* being used in the matching algorithm, but rather the raw log magnitude data of the Mel-spectrograms is used.

A section of speech with a single speaker was selected from the training set as a reference. It was selected for its clarity and lack of distortion. The reference segment is longer than any source data initially, then shortened to be the same size before histogram matching. For this work, typical reference lengths are around ten seconds long. Each piece of test data is then histogram matched to shortened versions of this same reference before decoding. It is also important to note that this research's choice of reference is not optimal, and there is no reason the reference segment cannot be arbitrarily long. In fact, the longer the reference segment the better, theoretically. With this in mind, it is important to recognize that the reference histogram can be precomputed as a lookup table for real-time implementation.

Looking at Figure 4, it is notable that the matched log Mel-spectrogram's spectral structure is retained from the source, but the plot has gained some of the dynamic range properties from the reference. Specifically, the darker regions of the source become even darker in the matched.[1]

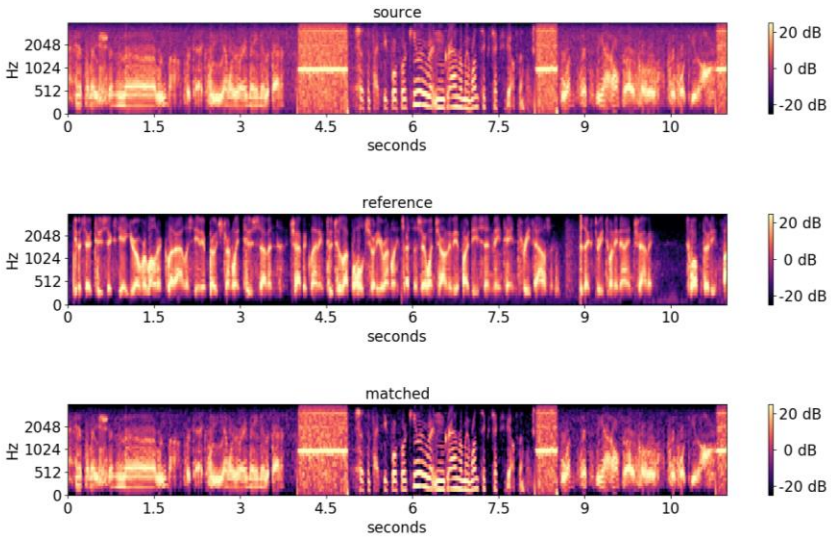

Figure 4: Naïve histogram matching approach

In Figure 5, the CDFs show that after histogram matching, the matched CDF aligns nicely to the reference CDF.

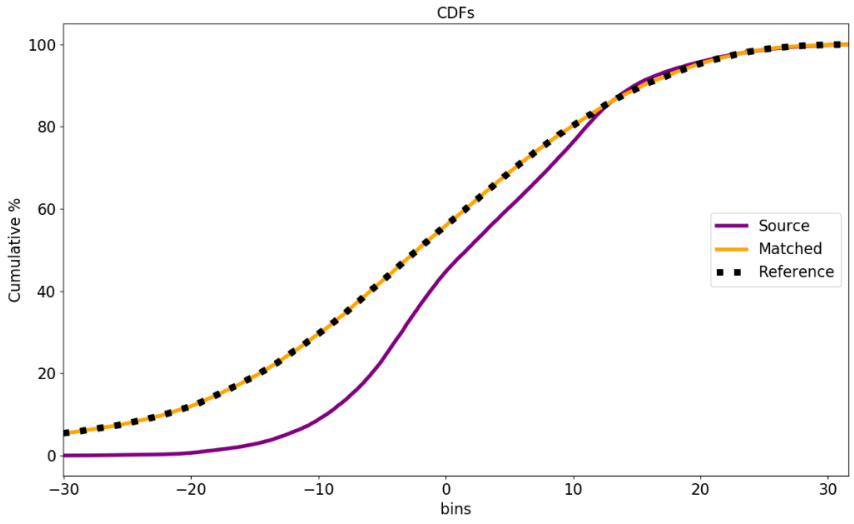

Figure 5: Naïve histogram matching CDFs

[1] Note that Figure 4 has a different magnitude normalization than Figures 6 and 8. This is due to Figure 4 being implemented as a preliminary test and proof of concept. Figures 6 and 8 are produced after the histogram matching algorithm is embedded into the ASR system; thus, they have a different magnitude normalization and silence added to the beginning of the source due to preprocessing in the ASR pipeline.

ii. Mel-frequency-dependent approach

Taking a leap further, if each Mel-frequency is treated as its own channel, and histogram matching is applied to corresponding frequencies between the source and reference log Mel-spectrograms, additional degrees of freedom can be added to the process. The difference between the naïve method and this one is that the histogram matching is at a finer scale and frequency dependent. In the same vein of thought, image processing splits data into color channels, whereas this method splits data into Mel-frequency channels. Figure 6 illustrates Mel-frequency-dependent histogram matching. Figure 7 shows that there are now multiple CDFs – one for each Mel-frequency, each plotted in a different color. In this case there are 40 discrete Mel-frequencies. Each source CDF is being matched to each reference CDF of the same Mel-frequency. Each Mel-frequency is redistributed differently – this is an indication that the frequency-dependence is a necessary facet to accurately reduce the acoustic mismatch between the source and the reference. From a visual standpoint the matched and reference now share the same slight high pass filtering around 256 Hz.

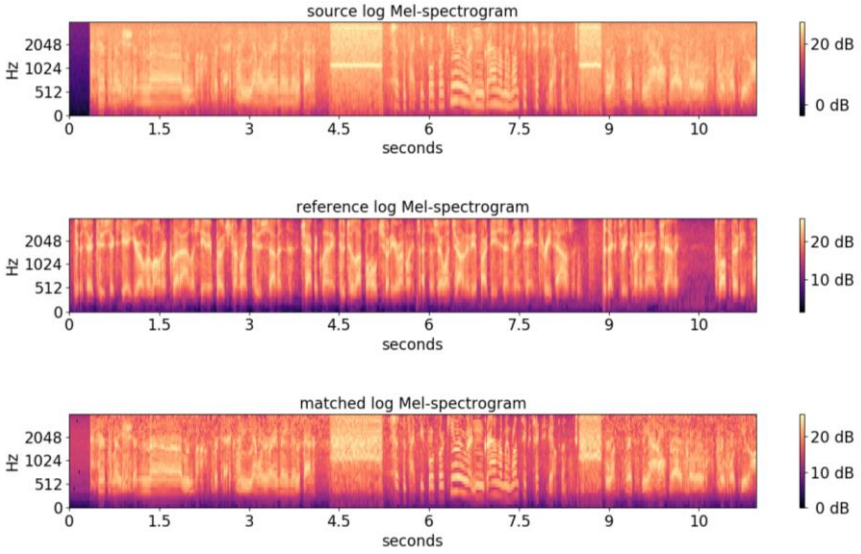

Figure 6: Mel-frequency-dependent histogram matching log Mel-spectrograms

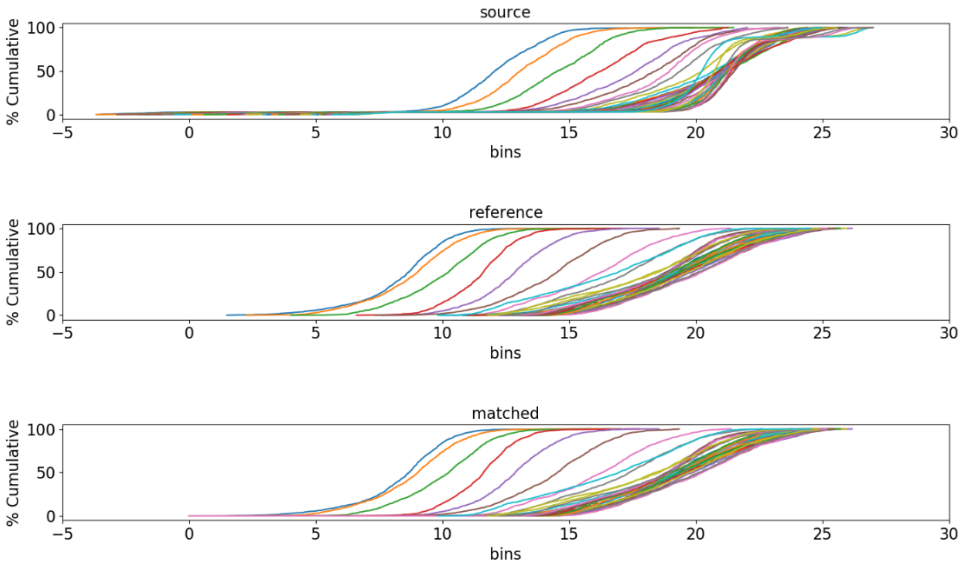

Figure 7: Mel-frequency-dependent histogram matching CDFs

iii. Mel-frequency-dependent approach with silence threshold

The final step is to introduce a silence threshold. Notice the silence at the very beginning of the source in Figure 6 has been redistributed to higher log magnitude values in the matched. This is undesirable as silence should remain silent as to not be mistaken for speech by the ASR system when decoding. One reason this undesirable silence redistribution occurs is that the reference contains very little silence. Therefore, there is no appropriate target quantization bin level for the source's silence to be matched to. This issue could be assuaged by choosing reference speech with some silence in it; however, it is more effective to introduce a silence threshold, as it is more quantitatively tunable and allows us to avoid the opposite problem where silence overly present. The silence threshold prevents the histogram matching from modifying values that correspond to silence. This silence threshold is manually and intuitively chosen.

Let it also be noted that at a low enough silence threshold, there is essentially no silence threshold, so the system can elect not to include the feature if the system performs better without it. Figure 8 shows the effects of Mel-frequency-dependent histogram matching with a silence threshold. Compare the former result from Figure 7 to Figure 9. Notice how the lower quantization level

around 10%, corresponding to silence in the source, is now retained in the CDFs of the matched when the silence threshold is incorporated.

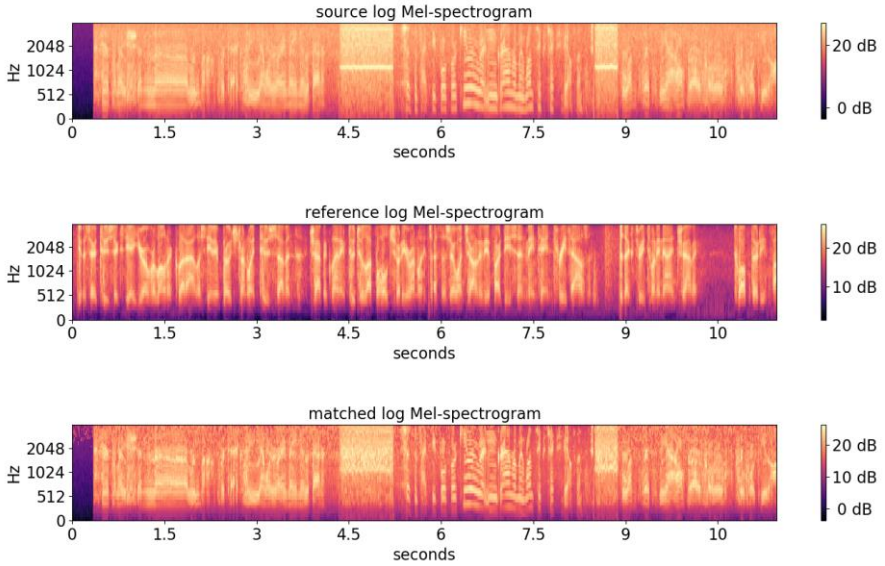

Figure 8: Mel-frequency-dependent histogram matching with silence threshold log Mel-spectrograms

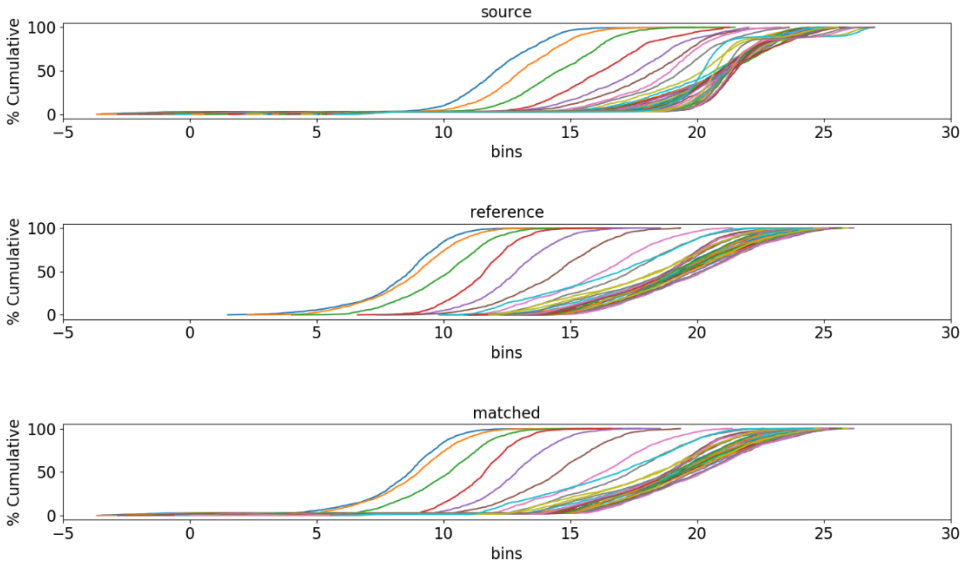

Figure 9: Mel-frequency-dependent histogram matching with silence threshold CDFs

1.4 Mathematic Overview

i. Building blocks

For a log Mel-spectrogram indexed in Mel-frequency by $\boldsymbol{i}$, and in time by $\boldsymbol{j}$,

$$\boldsymbol{S}[i, j],$$

The histogram of each Mel-frequency is estimated by quantizing the continuous log magnitude Mel-spectrogram values into discrete bins using some bin tolerance value, $\boldsymbol{\epsilon}$. The Mel-frequency-dependent histogram, $\boldsymbol{H}[\boldsymbol{i}, \boldsymbol{k}]$, is indexed in Mel-frequency by $\boldsymbol{i}$ and in quantization level by $\boldsymbol{k}$. There are $\boldsymbol{N_i}$ quantization levels per Mel-frequency, and $\boldsymbol{M}$ is the number of discrete Mel-frequencies. Figure 10 provides a visualization.

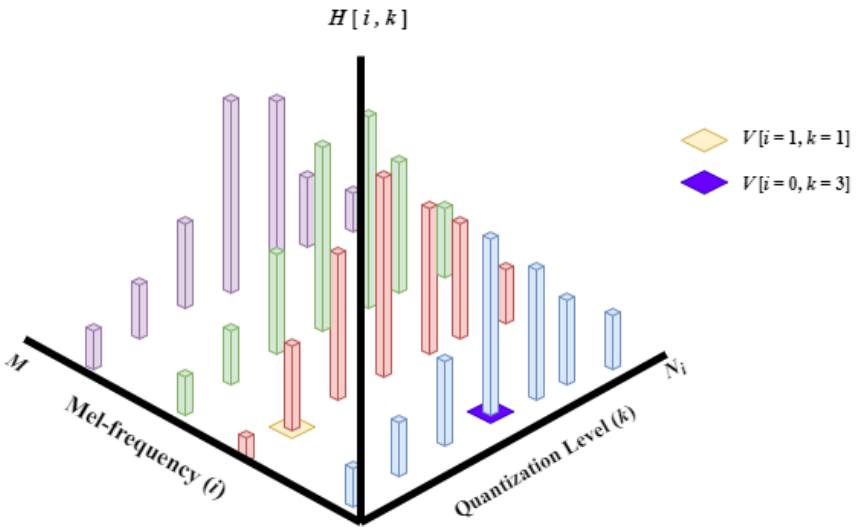

Figure 10: Visualization of a Mel-frequency-dependent histogram $\boldsymbol{H}[i, k]$. Not to scale, merely a toy example.

$\boldsymbol{V}[\boldsymbol{i}, \boldsymbol{k}]$, the log magnitude that specifies each quantization level, is indexed in Mel-frequency by $\boldsymbol{i}$ and in quantization level by $\boldsymbol{k}$. The time indices at which each quantization level, $\boldsymbol{V}[\boldsymbol{i}, \boldsymbol{k}]$, occurs in $\boldsymbol{S}[\boldsymbol{i}, \boldsymbol{j}]$ are given by $\boldsymbol{I}[\boldsymbol{i}, \boldsymbol{k}, \boldsymbol{l}]$, where $\boldsymbol{l}$ indexes the time values, $\boldsymbol{j}$, at which $\boldsymbol{S}[\boldsymbol{i}, \boldsymbol{j}] = \boldsymbol{V}[\boldsymbol{i}, \boldsymbol{k}]$. More simply, $\boldsymbol{I}[\boldsymbol{i}, \boldsymbol{k}, \boldsymbol{l}]$, remembers where each value in the histogram came from in the Mel-spectrogram so that the histogram can be inverted.

The Mel-frequency-dependent cumulative histogram $\boldsymbol{C'[i,k]}$ is calculated by the cumulative sum of each Mel-frequency histogram,

$$\boldsymbol{C'}[i,k] = \sum_{k=0}^{N_i-1} \boldsymbol{H}[i,k]\,, for\ i = 0, \dots, M-1,$$

Each cumulative histogram, $\boldsymbol{C'[i\,,k]}$, is normalized by its maximum (last) value to produce the Mel-frequency-dependent CDF $\boldsymbol{C[i,k]}$**.** See Figure 11.

$$\boldsymbol{C}[\,i,k\,] = \frac{\boldsymbol{C'}[i,k]}{\boldsymbol{C'}[i,N_i-1]}\,, for\ i = 0, \dots, M-1; for\ k = 0, \dots, N_i-1,$$

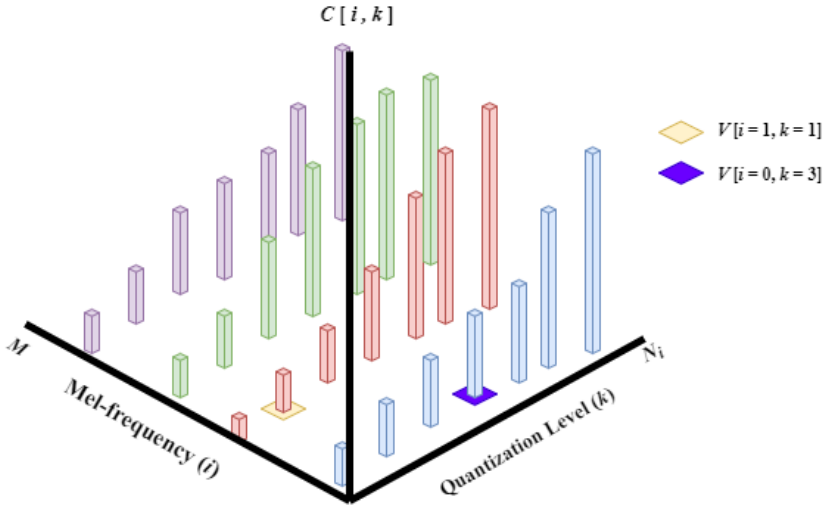

Figure 11: Visualization of Mel-frequency-dependent cumulative distribution function $\boldsymbol{C}[i,k]$. Not to scale, merely a toy example.

In order to perform histogram matching we need to define a linear interpolation function, $\boldsymbol{L}$. This is done by solving point-slope form for $\boldsymbol{x}$, where $\boldsymbol{m}$ is the slope between two points in the CDF of the reference.

$$y - y_1 = m(x - x_1),$$

$$m = \frac{(y_2 - y_1)}{(x_2 - x_1)},$$

$$x = \boldsymbol{L}(\,y\,,\,y_2\,,\,y_1\,,\,x_2\,,\,x_1\,) = \frac{(y - y_1)(x_2 - x_1)}{(y_2 - y_1)} + x_1,$$

ii. Mel-frequency-dependent histogram matching

Now that the building blocks are defined, the histogram matching begins with the linear interpolation between $\boldsymbol{C_{src}[i, k_{src}]}$ and $\boldsymbol{C_{ref}[i, k_{ref}]}$ (corresponding to $\boldsymbol{S_{src}[i, j_{src}]}$ and $\boldsymbol{S_{ref}[i, j_{ref}]}$). Matched quantization levels, $\boldsymbol{V_{mtc}[i, k_{src}]}$, are produced by the linear interpolation. Figure 12 gives a visual description of the interpolation process. When a source CDF value lies between two reference CDF values, a line is draw between the two reference CDF points and the source CDF value is projected from the y-axis onto the drawn line. Then, projecting down onto the x-axis gives the new quantization level.

$$\boldsymbol{V_{mtc}}[i\,, k_{src}] = \boldsymbol{L}\big(\boldsymbol{C_{src}}[i, k_{src}], \boldsymbol{C_{ref}}[i\,, k_{ref}], \boldsymbol{C_{ref}}[i\,, k_{ref}+1]\,, \boldsymbol{V_{ref}}[i, k_{ref}], \boldsymbol{V_{ref}}[i, k_{ref}+1]\big),$$

$$for\ i = 0, \dots, M-1; for\ k_{src} = 0, \dots, N_{i,src} - 1;\ for\ k_{ref} = 0, \dots, N_{i,ref} - 1;$$

$$where\ \boldsymbol{C_{ref}}[i\,, k_{ref}] \leq\ \boldsymbol{C_{src}}[i\,, k_{src}] \leq \boldsymbol{C_{ref}}[i\,, k_{ref}+1],$$

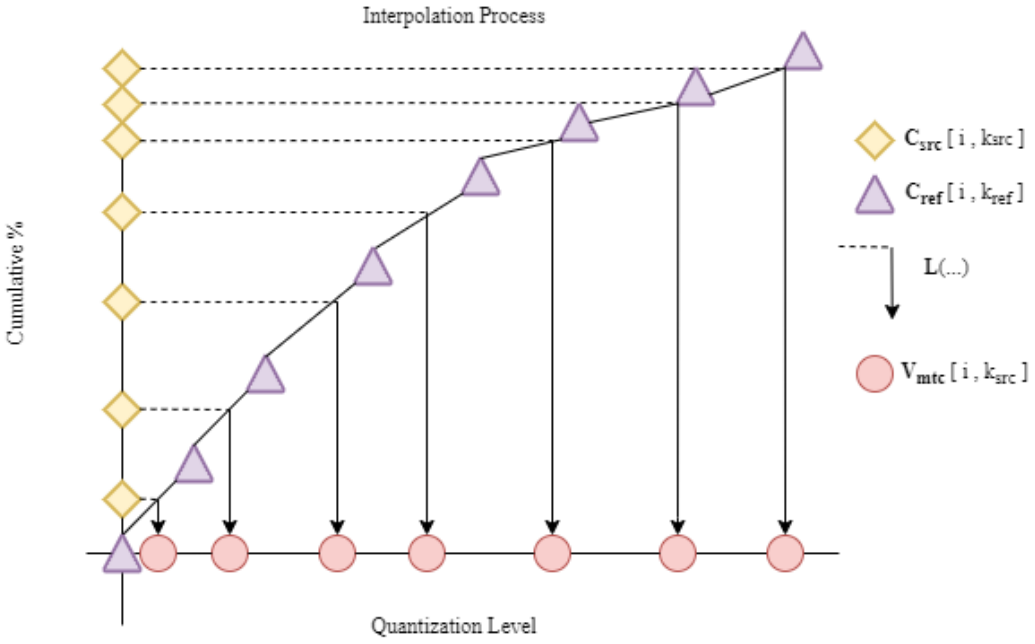

Figure 12: Interpolation visualization

Finally, the log Mel-spectrogram quantization level, $\boldsymbol{V_{src}}[i, k_{src}]$ is replaced by the histogram matched quantization level, $\boldsymbol{V_{mtc}}[i, k_{src}]$, using $\boldsymbol{I_{src}}[i, k_{src}, l]$ as a key. As a final condition, the matched quantization level only replaces the old quantization level if the original level is above some silence threshold, $\boldsymbol{\theta}$. As per,

$$\boldsymbol{S_{src}}\left[\, i\,, \boldsymbol{I_{src}}[\, i\,, k_{src}, l]\right] = \begin{cases} \boldsymbol{V_{mtc}}[\, i\,, k_{src}], & \boldsymbol{V_{src}}[i, k_{src}] > \theta \\ \boldsymbol{V_{src}}[\, i\,, k_{src}], & \boldsymbol{V_{src}}[i, k_{src}] \leq \theta' \end{cases}$$

$$for\ i = 0, \dots, M-1; for\ k_{src} = 0, \dots, N_{i,src} - 1; for\ l = 0, \dots, \boldsymbol{C_{src}}[i, k_{src}] - 1,$$

See Appendix A for a pseudocode description of the algorithm and its functions.

Chapter 2
METHODOLOGY

2.1 Data

This research was inspired by the study of air traffic control communications between airplane pilots and the control tower operators at various airports. Specifically, the data is coming from two sets. The first set is a very small (low-resource) dataset that contains 341 recordings with an average length of about ten seconds from 4 distinct airports (Renton, Everett, Moses Lake, Boston). This data will be referred to as "GND" data to abbreviate "ground communications." This is labelled data with near perfect transcriptions. Each transcription was captured by an experienced pilot. In addition to this dataset being small, it also contains a few other constraints. Firstly, the GND data is severely compressed. Figure 13 illustrates the limited dynamic range of a handful of GND recordings. This compression and distortion are caused by gain settings on the recording devices placed inside the airports.

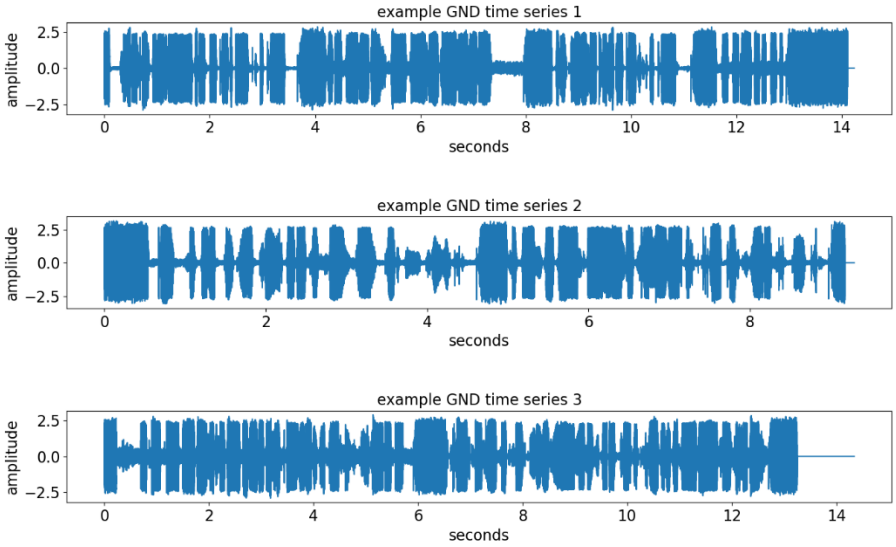

Figure 13: Three example GND files illustrating the level of compression and distortion.

The GND data is also very noisy. Figure 14 shows the presence of narrowband tonal noise occasionally occurring around 1024 Hz as well as broadband automatic gain control (AGC) noise present in between speech. The AGC noise usually appears at the beginning and end of a recording and occasionally between speaker changes. In general, recordings from Renton and Everett are the most noisy, distorted, and compressed.

The GND data is split up into three sets: training, validation, and test – as is commonly done in machine learning applications. The training set intentionally does not include one of the 4 airports (Renton) so that the system is configured to test for airport generalization. The validation set is used to fine-tune the ASR system, and the test set is used to evaluate how well the system generalizes to never-before-seen data. Table 1 shows how the data was divided.

	Train	Validation	Test	Totals:
Renton	0	42	41	83
Everett	46	20	40	106
Moses Lake	109	20	20	149
Boston	3	0	0	3
Totals:	158	82	101	341

Table 1: Data training, validation, and test splits.

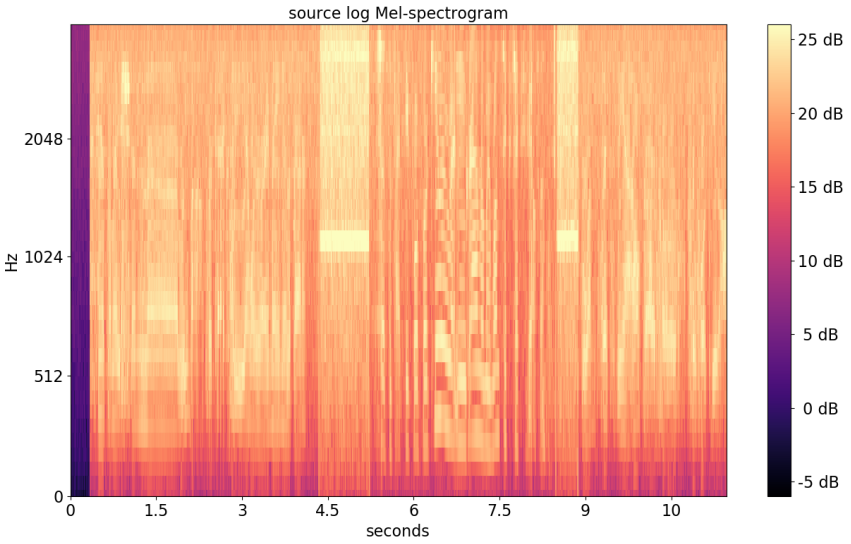

Figure 14: Typical narrow band tonal noise and wideband AGC noise present at time points 4.6 and 8.75 seconds

The second set of data was acquired from the Linguistic Data Consortium [2]. It will be referred to as “LDC” data. This data is very similar to the GND data as it also contains air traffic control communications between airplane pilots and the control tower. This data is much more abundant and was also transcribed by experienced pilots. There is around 70 hours of audio split into 41 recordings from 3 different airports. This data is generally less noisy, less compressed, and less distorted than the GND data. Figure 15 shows a typical time series from the LDC dataset.

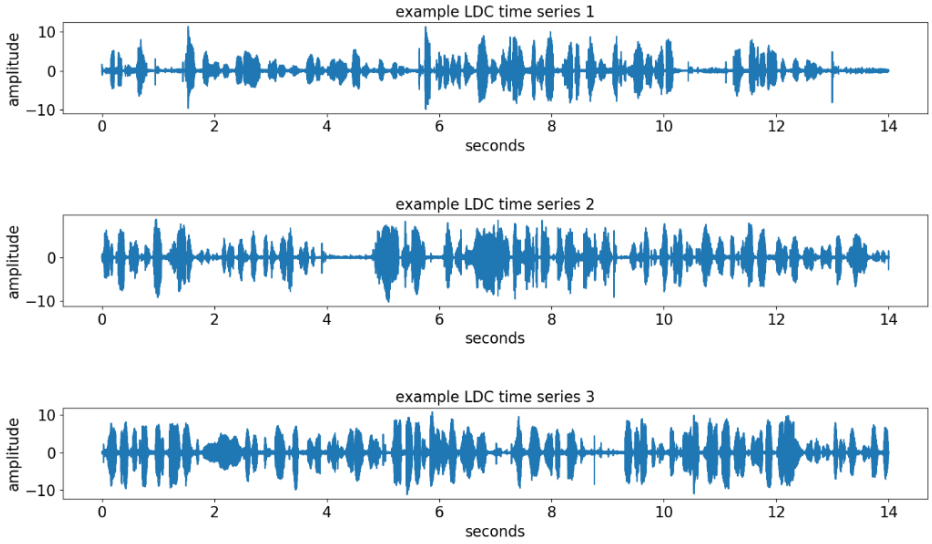

Figure 15: Typical LDC time series

Both datasets have a sampling rate of 8000Hz; their average spectrums are shown in Figure 16. Note the tonal spike around 1000 Hz in the GND data appearing between the LDC's main peaks. This quite possibly corresponds to the tonal noise often present in the GND data. The LDC data also has a slight high pass filtering around 250 Hz and a much faster high frequency roll off than the GND data.

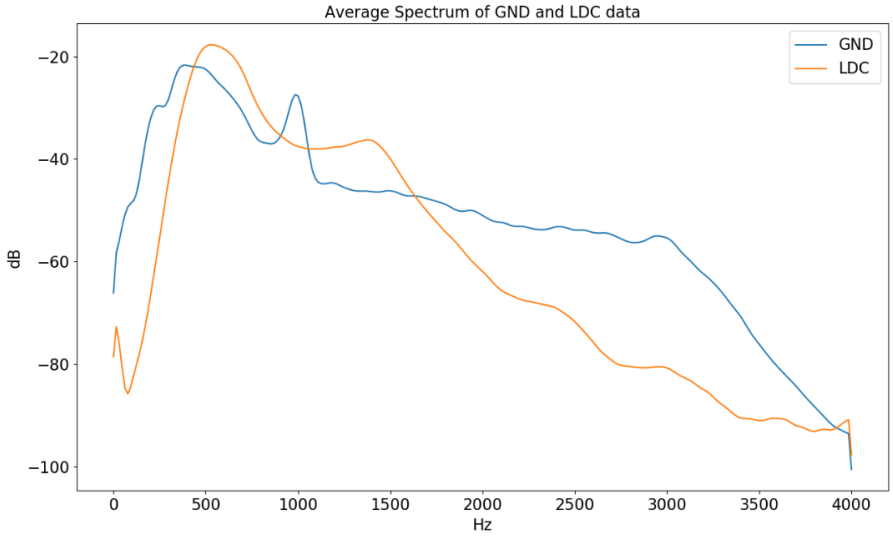

Figure 16: Plots obtained by averaging many GND and LDC Welch spectrums [5].

2.2 Experiment Architecture

The main goal of the research is to improve an ASR system built to automatically transcribe air traffic control conversations to reduce the amount of aircraft miscommunication. The system is tailored to recognize speech in the GND dataset and generalize well between airports, as the GND data is most relevant to the funders of this research. To best understand how the log Mel-spectrogram histogram matching was implemented and appreciate how it was tested, it is necessary to first know the details of the ASR system in which it was embedded.

The ASR system was implemented using Kaldi, an online open-source speech processing toolkit written primarily in C++ [9]. The ASR paradigm is implemented via the Kaldi nnet3 TDNN-F recipe [10] and is comprised of a time-delay neural network acoustic model and a trigram language model. The acoustic model uses a data augmentation method called SpecAugment. This modifies the log Mel-spectrogram of random recordings by masking certain sections of frequency and time (an example is shown in Figure 17). This data augmentation method is used to prevent overfitting and increase the performance of a neural net [3]. Trigram language models are used to aid word prediction by providing the probability of a word occurring conditioned on the two previous words.

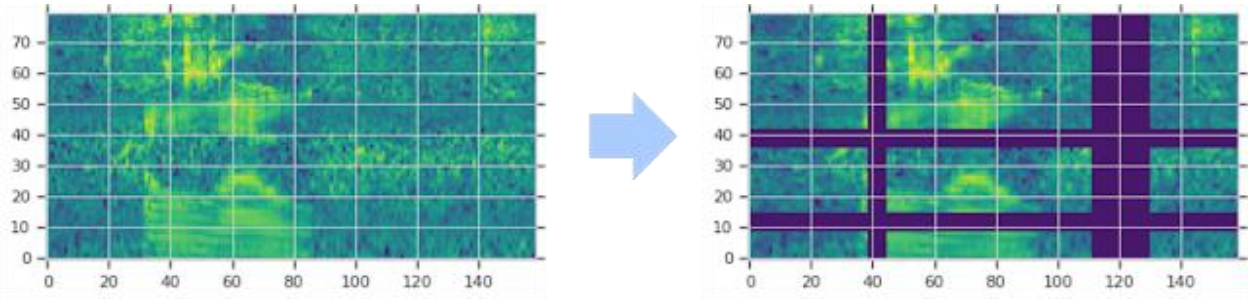

Figure 17: SpecAugment Example
https://ai.googleblog.com/2019/04/specaugment-new-data-augmentation.html

Kaldi nnet3 provides tools to build a hybrid system combining an acoustic model and a language model. The acoustic model is trained using only LDC data. The language model is trained using LDC data and 10 GND recordings from the entire GND set. See Figure 18 for a high-level system diagram.

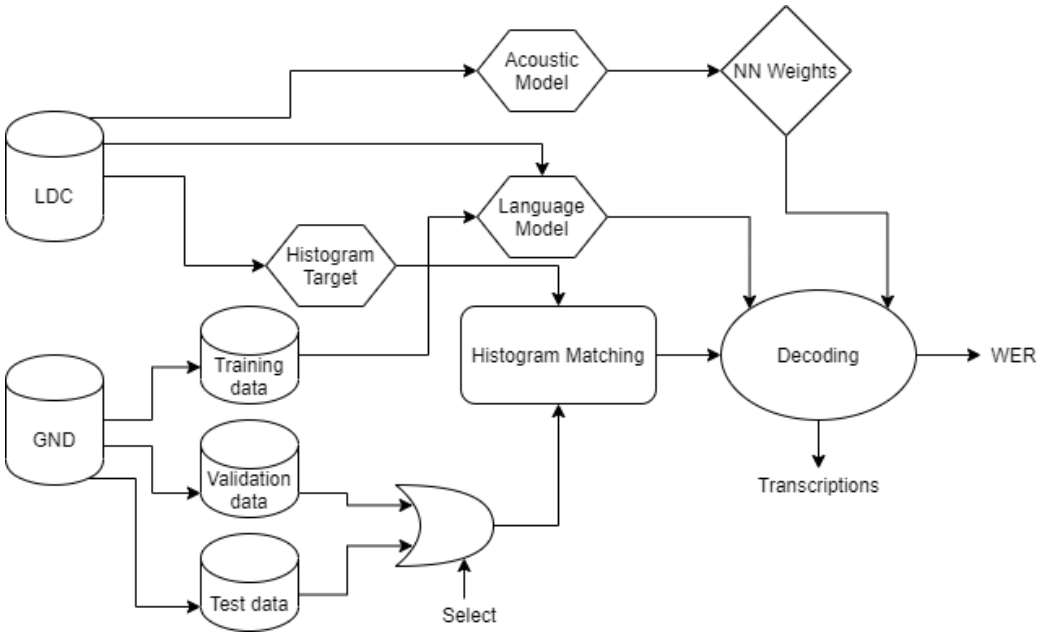

Figure 18: Diagram of how data is organized, split up for experiments, and how histogram matching is incorporated into the pipeline before decoding

There are a handful of tunable parameters within the histogram matching process. First it is necessary to choose how many discrete Mel-frequencies, $\boldsymbol{M}$, will be used in this process. For our purposes we used 40 Mel-frequencies for high resolution in the Mel-spectrogram. The low cutoff for these Mel-frequency bins was 20 Hz and the high cutoff was 3600 Hz for our 8000 Hz sampling rate. The tolerance value, $\boldsymbol{\epsilon}$, used to discretize the continuous values of the log Mel-spectrogram into quantization levels determines the number of quantization levels per Mel-frequency, $\boldsymbol{N_i}$. For our purposes we used a value of 10E-7 for $\boldsymbol{\epsilon}$. This value showed the best performance in terms of our performance metrics (which are introduced in Chapter 3) when compared to other nearby tolerance values in the range [10E-2, 10E-9]. The process of choosing the silence threshold, $\boldsymbol{\theta}$, was simple but time consuming. Iterating through a reasonable neighborhood of values at a small step size can reveal effective threshold values. As noted earlier, it is preferable to keep the silence threshold relatively low as to avoid solutions where the speech can become only partially histogram matched - possibly hindering its ability to generalize well. The final tunable parameter

is the choice of reference speech used in histogram matching. Figure 18 calls this the “Histogram target.” As stated in Section 1.3, this research used a single segment of speech from one speaker as reference for every histogram matching instance. The reference segment is made the same length as each source segment. This is not likely the optimal choice as. Theoretically, the longer the segment of reference data that is put into the reference histogram, the more accurate the reference log magnitude probability distribution will be. Additionally, this way the reference histogram can be precomputed for efficiency.

Chapter 3
EXPERIMENTS AND RESULTS

In order to test the effectiveness of histogram matching with a silence threshold, the National Institute of Standards and Technology (NIST) Word Error Rate (WER) is used to compare the baseline ASR system performance to the ASR system performance after Mel-frequency-dependent histogram matching is incorporated [8]. As a second measure, a bootstrap paired comparison is used for significance tests [7]. These will be our performance metrics to compare our results.

3.1 Validation Set

In Figure 19, the baseline WER on the validation dataset is plotted in red. The blue line is a plot of the WER with histogram matching as a function of the silence threshold on the validation dataset.

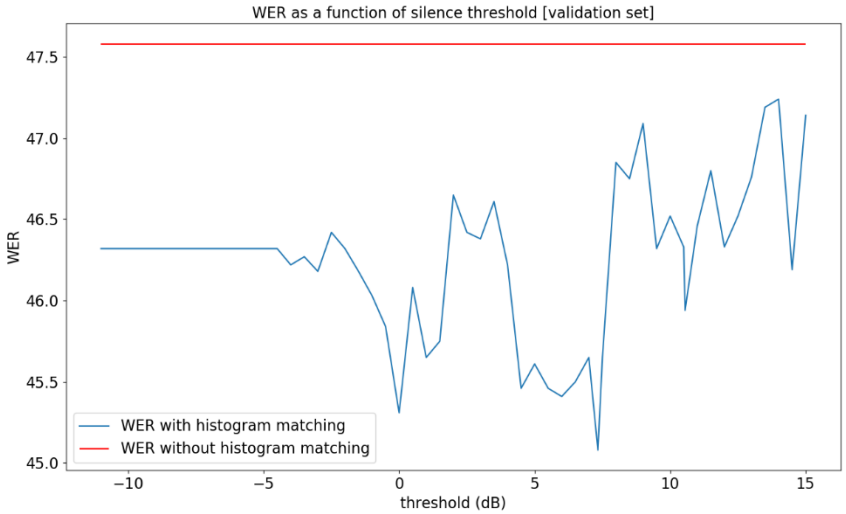

Figure 19: WER as a function of silence threshold for the validation dataset

Figure 20 shows chance of confidence improvement as a function of silence threshold.

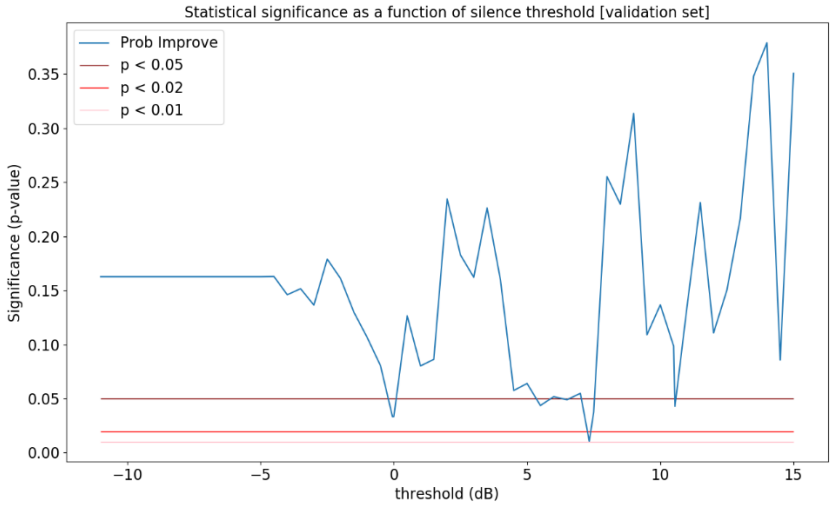

Figure 20: Bootstrap paired comparison test as function of silence threshold for the validation dataset

Figure 21 shows three promising regions that, on average, have the lowest p-values coupled with lowest WERs. A value from each of these three regions is chosen for use on the test set because the regions represent seemingly separate solutions and are remote from one another.

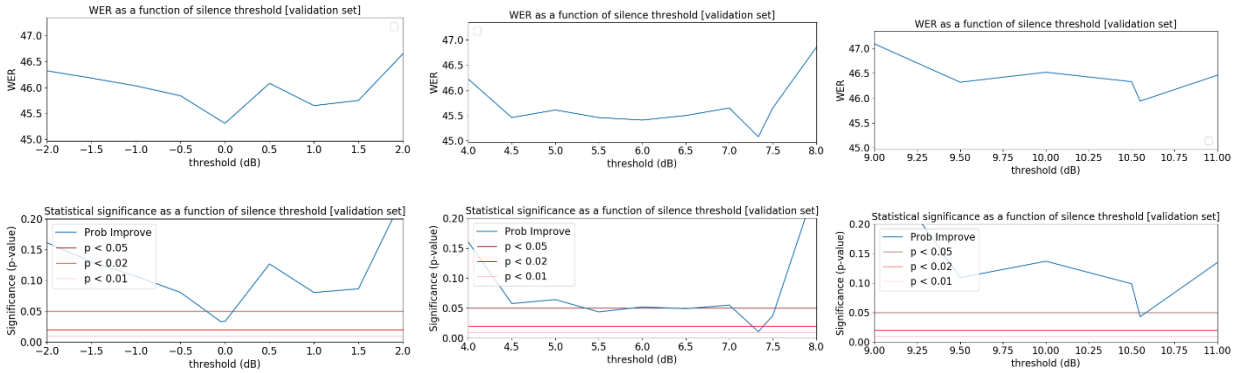

Figure 21: From left to right Regions 1, 2, and 3 containing effective silence threshold values.

Table 2 shows the results from using optimal silence threshold values from Region 1, 2, and 3 on the validation set. Table 2 also shows the results per airport.

	WER (Baseline = 47.6%)	Significance	WER per Airport (Baseline)	Significance per Airport
			Renton 51.6% Everett 40.3% Moses Lake 42.6%	
Region 1 (threshold=-0.05 dB)	45.4%	$p < 0.04$	Renton 50.2% Everett 37.9% Moses Lake 37.5%	Renton $p < 0.22$ Everett $p < 0.11$ Moses Lake $p < 0.04$
Region 2 (threshold=7.334 dB)	45.1%	$p < 0.02$	Renton 49.1% Everett 39.7% Moses Lake 37.4%	Renton $p < 0.05$ Everett $p < 0.38$ Moses Lake $p < 0.03$
Region 3 (threshold=10.55 dB)	46.0%	$p < 0.05$	Renton 50.4% Everett 39.3% Moses Lake 38.6%	Renton $p < 0.20$ Everett $p < 0.28$ Moses Lake $p < 0.03$

Table 2: A comparison of optimal silence thresholds on the validation set

3.2 Test Set

Table 3 shows the results from using the three optimal silence thresholds from the validation set on the test set.

	WER (Baseline = 50.4%)	Significance	WER per Airport (Baseline)	Significance per Airport
			Renton 58.0% Everett 49.1% Moses Lake 29.9%	
Region 1 (threshold=-0.05 dB)	46.8%	$p < 0.001$	Renton 54.1% Everett 44.2% Moses Lake 30.4%	Renton $p < 0.007$ Everett $p < 0.009$ Moses Lake Not Significant
Region 2 (threshold=7.334 dB)	50.9%	Not Significant	Renton 54.4% Everett 53.5% Moses Lake 33.0%	Renton $p < 0.005$ Everett Not Significant Moses Lake Not Significant
Region 3 (threshold=10.55 dB)	51.2%	Not Significant	Renton 55.6% Everett 53.3% Moses Lake 32.2%	Renton $p < 0.030$ Everett Not Significant Moses Lake Not Significant

Table 3: WER and bootstrap paired comparison test results for test set using optimal threshold values from the validation set

In Figure 22 and 23, the WER and chance of confidence improvement results for the test set are plotted with all reasonable silence threshold values.

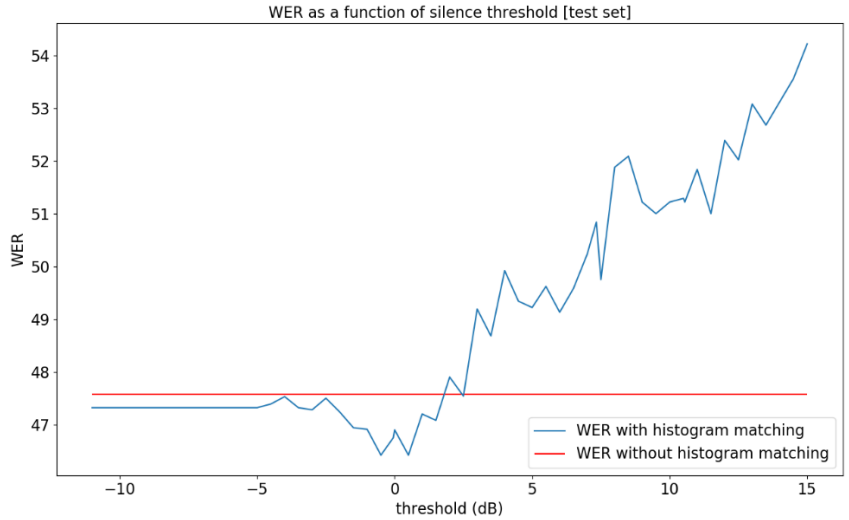

Figure 22: WER as a function of silence threshold while decoding with the test set.

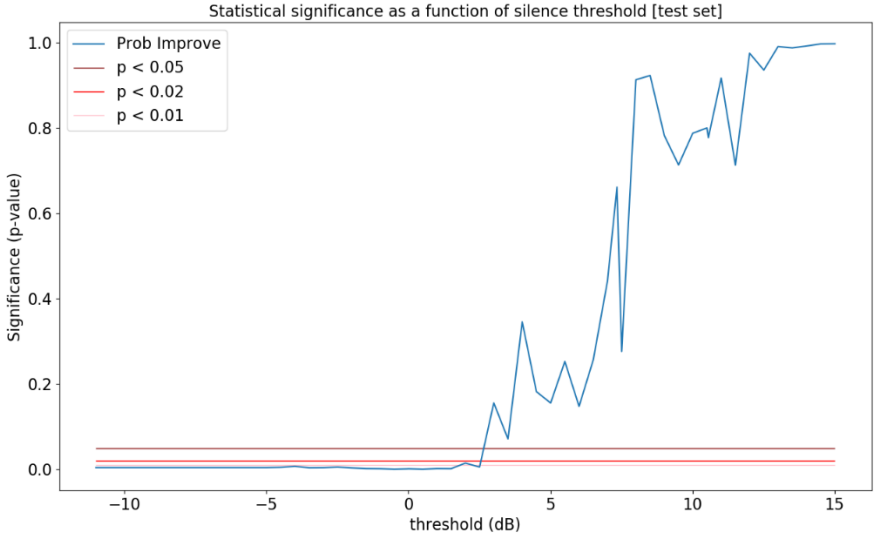

Figure 23: Bootstrap paired comparison test as a function of silence threshold when decoding with the test set.

Contrary to the validation dataset, the test set consistently prefers a lower threshold. Figure 24 illustrates a histogram matching example from the test dataset.

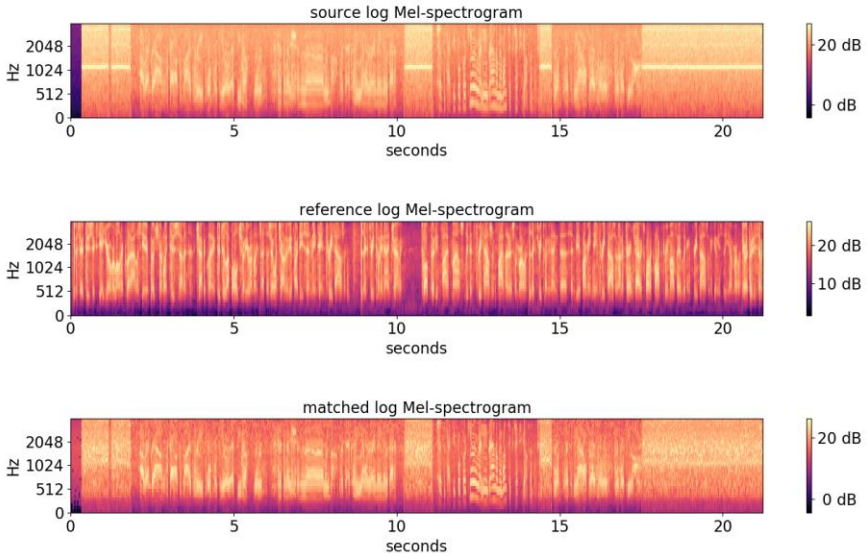

Figure 24: An example of Mel-frequency-dependent histogram matching on the test dataset with a silence threshold of -0.05 dB

Chapter 4
DISCUSSION

4.1 Analysis of Results

Contrasting the test and validation set, only one of the three effective threshold values chosen from the validation set is effective on the test set (-0.05 dB). The histogram matching has the best result on the test set with significance $p < 0.001$ using a validation-tuned threshold; however, it is important to address why only one threshold value generalized well. As mentioned prior, the GND dataset is very small and thus prone to high acoustic variance. Ideally the validation and test sets would have similar distributions, but this does not seem to be the case. The test set contains 20 more files from a specific airport than the validation set; otherwise, they have the same amount of data from each airport. Because of the high amount of variance in the datasets, it makes sense that a silence threshold value around or below 0 dB would work best because any silence threshold above 0 dB starts to creep into the region in which speech is present. This is not likely to produce a stable solution as the speech will be modified (partially histogram matched) and might produce highly variable results across speech recordings as it interacts with the neural net weights. This reflects perhaps a poor solution where the silence threshold is doing more than it was intended to do. Looking back at Figures 19, 20, 22, and 23, shows that silence threshold values around or below 0 dB present more stable solutions than values above 0 dB.

If we look at the airport-specific results, we can see that for the validation set Region 1 provides the most stable bootstrap and WER results, whereas Region 2 and 3's results vary more across airports. This stability is reflected in the test set as Region 1 is the only region that positively effects both Renton and Everett. Interestingly, the results for Moses Lake were never significant

in the test set. This could be because Moses Lake is the least noisy, but Moses Lake's performance in the validation set counters this theory and illustrates that the dataset is indeed very small and prone to issues due to high acoustic variance.

4.2 Acoustic Effects

One of the most interesting effects of histogram matching is its effect on tonal noise. Examining Figure 25, a specific speech recording that contains a tonal noise present within AGC broadband noise, one can see that the tonal noise is redistributed or squashed into multiple Mel-frequencies making it more similar to broadband noise. This makes regions such as this less harmonic and less correlated with speech. Perhaps this allows the ASR system to better identify/ignore these regions as noise. As mentioned previously, it can also be seen that the slight high-pass filtering in the reference becomes present in the matched log Mel-spectrogram after histogram matching.

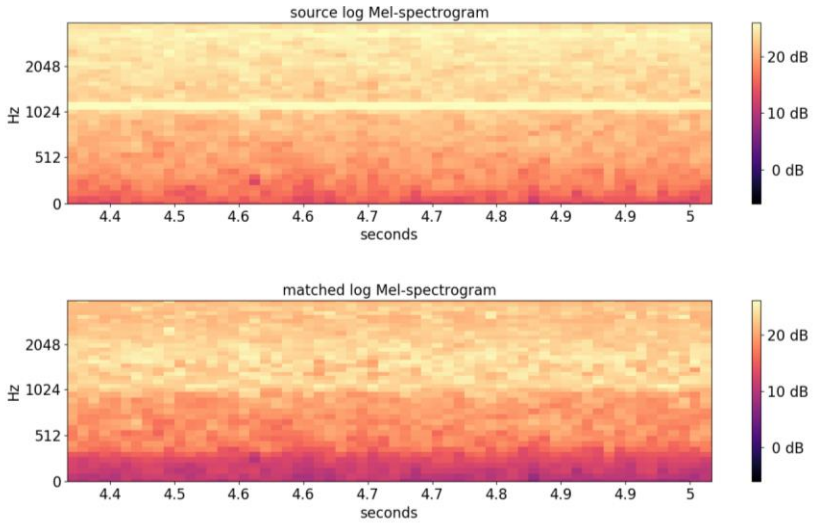

Figure 25: Zoomed log Mel-spectrogram from Figure 8 to illustrate the effect of Mel-frequency-dependent histogram matching with silence threshold on tonal noises.

Chapter 5

CONCLUSIONS AND FUTURE WORK

Using Mel-frequency-dependent histogram matching with a near-optimal silence threshold, we were able to lower the WER from 47.6% to 45.4% with a significance of $p < 0.04$ in the validation set. In the test set, using the same silence threshold, we were able to lower the WER from 50.4% to 46.8% with a significance of $p < 0.001$. Mel-frequency-dependent histogram matching has proved itself an effective tool to deal with acoustic mismatch. The following paragraphs will suggest directions for future work.

There are several ways the current approach could be optimized. A more complex interpolation algorithm such as a 5-point Sinc interpolation could replace the linear interpolation. Sinc interpolation always outperforms linear interpolation; however, it remains unclear whether the log Mel-spectrogram's CDFs would benefit from a more accurate interpolation. Additionally, the processing of choosing the silence threshold could be made automatic and airport dependent. This could increase the performance and decrease the amount of time spent tuning parameters. As stated before, it would also be beneficial to test this histogram matching approach with a larger amount of data used to build the reference histogram. Correspondingly, one could choose the reference based on where the training and validation/test data differ the most in terms of energy (or some other metric) and cross reference those sections with sections where the system performs best on the training data. In this way, foreign sections might be more easily recognizable to the ASR system after they are matched to sections of speech where the ASR system performs well.

The complexity of the system could be increased by making the histogram matching time-varying. One could apply the histogram matching differently to steady state sections versus transient sections of speech or perhaps modify certain phonemes differently. In a different vein, histogram

matching could be used as a data augmentation method. One way this could be implemented is by histogram matching random training data to reference data from the validation or test set within the training process, then evaluate using WER.

In a broader sense, histogram matching could be used for other problems and applications. One could use histogram matching on the STFT of a speech segment and attempt to invert the STFT afterwards and listen to the audio. This inversion would not be possible with the log Mel-spectrogram because the phase is thrown out when the logarithm is taken. This might be useful in musical effects, speech enhancement, or speaker identification. In higher dimensional problems, histogram matching could be used on the log Mel-spectrogram of images. If implemented similarly, histogram matching could, in simple terms, align the visual characteristics of image edges (high frequencies) and image smoothness (low frequencies). This could possibly benefit image recognition systems.

REFERENCES

[1] J. Xu *et al.*, "LRSpeech: Extremely low-resource speech synthesis and recognition," *arXiv [eess.AS]*, 2020.

[2] "Air Traffic Control Complete," *Upenn.edu*. [Online]. Available: https://catalog.ldc.upenn.edu/LDC94S14A. [Accessed: 12-Nov-2020].

[3] D. S. Park and W. Chan, "SpecAugment," *Googleblog.com*, 22-Apr-2019. [Online]. Available: https://ai.googleblog.com/2019/04/specaugment-new-data-augmentation.html. [Accessed: 12-Nov-2020].

[4] "Histogram matching — skimage v0.18.0.dev0 docs," *Scikit-image.org*. [Online]. Available: https://scikit-image.org/docs/dev/auto_examples/color_exposure/plot_histogram_matching.html. [Accessed: 12-Nov-2020].

[5] P. Welch, "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE trans. audio electroacoust.*, vol. 15, no. 2, pp. 70–73, 1967.

[6] R. C. Gonzales and B. A. Fittes, "Gray-level transformations for interactive image enhancement," *Mech. Mach. Theory*, vol. 12, no. 1, pp. 111–122, 1977.

[7] B. Efron, "Bootstrap methods: Another look at the jackknife," in *Springer Series in Statistics*, New York, NY: Springer New York, 1992, pp. 569–593.

[8] "Tools," *Nist.gov*. [Online]. Available: https://www.nist.gov/itl/iad/mig/tools. [Accessed: 12-Nov-2020].

[9] D. Povey, "Kaldi," *Kaldi-asr.org*. [Online]. Available: https://kaldi-asr.org/doc/index.html. [Accessed: 12-Nov-2020].

[10] D. Povey *et al.*, "Semi-orthogonal low-rank matrix factorization for deep neural networks," *Danielpovey.com*. [Online]. Available: https://www.danielpovey.com/files/2018_interspeech_tdnnf.pdf. [Accessed: 20-Nov-2020].

[11] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C–23, no. 1, pp. 90–93, 1974.

PROJECT FILES AND CODE

https://gitlab.com/cuinnfey/frequency_dependent_histogram_matching_with_silence_threshold

APPENDIX A

Algorithm

i. Helper functions

Algorithm 1 Cumulative Sum

Input: *counts* ▷ takes histogram bin counts
Output: *cumulative* ▷ returns cumulative histogram

$N \leftarrow length(counts)$
$cumulative[0] \leftarrow counts[0]$
for $k \leftarrow 1$ to $N-1$ **do**
    $cumulative[k] \leftarrow cumulative[k-1] + counts[k]$

Algorithm 2 Histogram Information

Inputs: S ▷ takes a log Mel-spectrogram frequency slice
Output: $values, counts, indices$

Initialize: $values, counts, indices$

$L = length(S)$
$\epsilon \leftarrow 10E-7$
$values \leftarrow sort(S)$ ▷ Sort low to high
$values \leftarrow unique(values, \epsilon)$ ▷ Eliminate duplicates with tolerance ϵ
$N \leftarrow length(values)$

for $k \leftarrow 0$ to $N-1$ **do**
    $tally \leftarrow 0$
    for $j \leftarrow 0$ to $L-1$ **do**
        $difference \leftarrow abs(values[k] - S[j])$
        if $difference <= \epsilon$ **then**
            $counts[k] \leftarrow counts[k] + 1$
            $indices[k][tally] \leftarrow j$
            $tally \leftarrow tally + 1$

Algorithm 3 Linearly Interpolate

Inputs: $y, x1, x2, y1, y2$
Output: x

$x = (((y - y1) * (x2 - x1))/(y2 - y1)) + x1$

Algorithm 4 Interpolate

Inputs: $CDF_{src}, values_{src}, CDF_{ref}, values_{ref}, \theta$ ▷ θ is the silence threshold
Output: $values_{mtc}$ ▷ returns new matched quantization levels

```
for i ← 0 to length(CDF_src) do
    for j ← 0 to length(CDF_ref) do
        if CDF_ref[j − 1] <= CDF_src[i] <= CDF_ref[j] then
            if CDF_src[i] >= θ then
                values_mtc[i] ← Linearly Interpolate(CDF_src[i],
↪                                                    values_ref[j − 1],
↪                                                    values_ref[j],
↪                                                    CDF_ref[j − 1],
↪                                                    CDF_ref[j])
            else
                values_mtc[i] ← values_src[i]
```

ii. Main algorithm

Algorithm 5 Histogram Matching

Inputs: S_{src}, S_{ref} ▷ takes a source and reference log Mel-spectrogram
Output: S_{mtc} ▷ returns the matched log Mel-spectrogram

$M, N = size(S_{src})$ ▷ Assume size of $S_{src} == S_{ref}$

Initialize S_{mtc} to be the same size as S_{src}

for $i \leftarrow 0$ to $M - 1$ **do** ▷ M is the number of discrete Mel-frequencies

$values_{src}, counts_{src}, indices_{src} \leftarrow Histogram\ Information(S_{src}[i])$

$values_{ref}, counts_{ref}, indices_{ref} \leftarrow Histogram\ Information(S_{ref}[i])$

$N_{i,src} \leftarrow length(values_{src})$
$N_{i,ref} \leftarrow length(values_{ref})$

$CDF_{src} \leftarrow Cumulative\ Sum(counts_{src})$
$CDF_{src} \leftarrow CDF_{src}/CDF_{src}[N_{i,src} - 1]$

$CDF_{ref} \leftarrow Cumulative\ Sum(counts_{ref})$
$CDF_{ref} \leftarrow CDF_{ref}/CDF_{ref}[N_{i,ref} - 1]$

$values_{mtc} \leftarrow Interpolate(CDF_{src}, values_{src}, CDF_{ref}, values_{ref}, values_{mtc}, \theta)$

for $k \leftarrow 0$ to $N_i - 1$ **do** ▷ replace old source values with new matched values
  for $l \leftarrow 0$ to $counts[k] - 1$ **do**
    $index \leftarrow indices_{src}[k][l]$
    $S_{mtc}[i][index] \leftarrow values_{mtc}[k]$

VITA

Cuinn Rios Fey received a bachelor's degree in Electrical and Computer Engineering in 2019 from the University of Washington. His research interests, partially inspired by his musicianship, include signal processing, machine learning, and automatic speech recognition in the context of hearing aids, audio engineering, music, images, and video.