Revealing Hidden Conversations in Privacy-Sensitive Audio Using Neural Networks

Pepijn Vunderink
Supervisors: Hayley Hung, Jose Vargas Quiros
EEMCS, Delft University of Technology, The Netherlands

June 19, 2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Abstract

With widespread use of advanced technology for the recording, storing and sharing of social interactions, protecting privacy of people has been a growing concern. This paper zooms in on the collection of spoken audio with regard for the privacy of recorded individuals.

Recently efforts have been made to collect audio at a low sampling rate to obfuscate spoken words in the recorded audio, such that conversations are kept private. This research investigates whether it is possible to upsample this low-resolution audio, using an existing super-resolution model, in order to reveal parts of the previously obfuscated conversations. The performance of the model is measured in terms of the word error rate of automatically generated transcriptions of the upsampled audio.

It turns out that it is possible to significantly increase the intelligibility of low-resolution privacy-sensitive audio by upsampling. Though the use of the super-resolution model seems to be limited when it comes to revealing significant parts of conversations.

1 Introduction

In a world with advanced technology for the recording, storing and sharing of social interactions, protecting privacy of people has been a growing concern. This is evidenced by a growing number of regulations concerning privacy [1]. Inevitably, this also has a great impact on how personal data is collected and distributed in all areas of scientific research. Consequently new studies are putting efforts into preserving the privacy of participants of data collections, while still being able to perform useful analyses on the collected data.

This paper zooms in on the collection of spoken audio with regard for the privacy of recorded individuals. Rhythm [2] and MINGLE [3] are examples of experiments where spoken audio was recorded and special attention was paid to the privacy of participants. The audio was recorded in such a way that spoken words are hard to understand, in an attempt to obfuscate the contents of conversations – this audio is called *privacy-sensitive audio*. This was achieved by recording the audio at a very low sampling rate, which results in loss of high frequency information. Many important speech cues, such as consonants, are typically contained within these higher frequency bands [4].

While these efforts are certainly promising, not a lot of research has been put into how sound the method of down-sampling is for the purpose of protecting privacy. There is also a lack of insight into how useful the remaining low sampling rate audio data is for performing analyses such as VAD (**V**oice **A**ctivity **D**etection), laughter detection or emotion detection.

In this paper I aim to address part of the aforementioned concerns, namely the soundness of down-sampling in protecting privacy. More specifically this paper is about exploring the possibility of hallucinating the – unrecorded – higher frequency components in order to increase the intelligibility of the recorded conversational audio and possibly recover hidden conversations. Hence, the main research question of this paper can be stated as: *Can existing super-resolution techniques be used to reveal hidden conversations in privacy-sensitive audio?*

Super-resolution is a synonym for bandwidth extension (BWE) – the term BWE is more commonly used in the world of audio processing. A lot of research has been done in the area of audio bandwidth extension [5]–[11]. Typically BWE techniques are meant to improve the quality of telephony speech or compressed music. However, BWE happens to do exactly what is needed for our purposes as well; infer (i.e. hallucinate) higher frequency components from low sampling rate audio data. I have used one of these existing models, a neural network designed by Kuleshov et al. [5], to explore whether it is possible to reveal conversations that were previously unintelligible.

After training, the model should be able to predict high-resolution audio – containing information in higher frequency bands – based on a low-resolution input signal. The model is then validated by transcribing predicted audio using automated speech transcription software and comparing the resulting hypothesis transcriptions to reference transcriptions by calculating the WER (**W**ord **E**rror **R**ate). By comparing the WER of the predicted audio transcription to the WER of a low-resolution audio transcription, some insight can be gained into whether the model is able to increase the intelligibility of spoken audio and in extension whether conversational information is revealed. Refer to section 3.3 for more information about WER and other metrics used in this paper.

The remainder of the paper is organized as follows. In section 2, some additional background is provided on the RhythmBadge and Midge, as well as some background on audio super-resolution. Section 3 describes the model and the datasets used for training evaluation, followed by . Section 4 presents the results, preceded by a precise description of the experimental setup. Section 5 reflects on ethical aspects and reproducibility of this research. Then finally in sections 6 and 7 the main conclusions are presented, followed by a discussion on the limitations of this research as well as some recommendations to what could be improved upon in follow-up research.

2 Background

This section provides the necessary foundations for understanding the main contributions of this paper. First more background is provided on the RhythmBadge and Midge experiments, which form the basis of this research. Then some background on audio super-resolution is given.

2.1 RhythmBadge

In 2018, the MIT Media Lab developed a measurement platform, called Rhythm, that was meant to aid research in the fields of computational social science and organizational design [2]. They also conducted a study around measuring face-to-face interactions in formal meetings, using the Rhythm platform. The study focused on hybrid meetings, meetings where some participants are in the same room and others are connected through an online video call. Using the RhythmBadge, a badge worn by each of the participants, three types of data were collected: vocal activity, inter-badge proximity and location (relative to beacons in the room). For privacy reasons the badge recorded audio at a low sample rate (700Hz). They found that this sampling rate was sufficient for their purposes, as the audio was only needed for VAD.

2.2 Midge

TUDelft's Socially Perceptive Computing lab carried out a data collection at the ConfLab conference in Nice [3]. Different kinds of data types were collected using a badge, called Midge, which was inspired by the design of the RhythmBadge. The collected data was meant for measuring and studying social interactions. Similarly to Rhythm, proximity and low sample rate audio were collected. Additionally, acceleration data was collected for measuring movements and gestures. A higher sampling rate of 1.25kHz was used for recording the audio, compared to the 700Hz adopted by the RhythmBadge. This is because the ConfLab setting – a busy networking event – came with a significant amount of additional noise (so-called

cocktail party noise), compared to the environment of the Rhythm experiment – a formal business meeting. It is yet to be verified whether the 1.25kHz sampling rate is adequate when it comes to privacy preservation.

2.3 Audio Super-Resolution

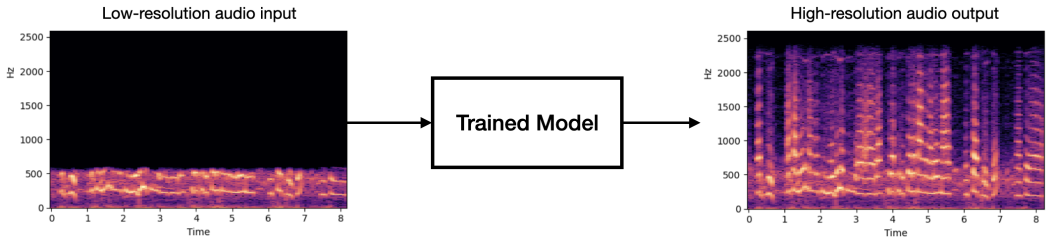

Figure 1: A visualization of audio super-resolution (also known as artificial bandwidth extension).

Audio super-resolution entails the inference of a high-resolution signal based on a low-resolution input signal. It can also be thought of as "hallucinating" higher frequencies that are missing in the low-resolution input. A visualisation of this process is shown in figure 1.

Methods for audio super-resolution, better known as (artificial) bandwidth extension – (A)BWE – in audio signal processing, have been widely studied – as summarized by Prasad and Kumar [12]. Over the years many different approaches have been developed for speech BWE. Usually these are developed with the goal of increasing the quality and intelligibility of speech over narrow-band (NB) telephone connections. According to Prasad and Kumar, speech BWE methods can be classified into two categories: model based techniques and non-model based techniques. In model based techniques a statistical model is derived from training data which is then used to predict a wide-band (WB) signal from a NB signal. The model can be as simple as a linear mapping, or more complex, such as the Gaussian Mixture Model or the Hidden Markov Model.

Recently the focus has shifted to models based on neural networks [5]–[11]. Methods using neural networks have proven to outperform previous methods and are often preferred over other methods because of the versatility of neural networks. A variety of neural network architectures have been proposed, where some operate on time domain audio features (based on the raw waveform of the audio) [5] and others on frequency domain features (e.g. spectrograms, achieved by first applying a Fast Fourier Transform to the waveform) [6], [7]. However, the best results so far have been achieved by taking advantage of a combination of both time and frequency domain features [8]–[11]. All of these hybrid solutions combine the time and frequency domains in a similar way: the networks operate on a time domain waveform both as input and output – this is commonly called an end-to-end solution –, and the loss function is some combination of a time domain loss measure and a frequency domain loss measure.

For this research I have chosen to use a super-resolution method based on a neural network, designed by Kuleshov et al. [5], which operates on audio in the time domain. Refer to section 3.1 for more information on this choice and the model itself.

3 Methodology

This section lays out the methodology that was adopted to answer the main research question. First I introduce the model that was used to perform super-resolution. Then I provide background on the datasets that were used to train and evaluate this model. And finally the evaluation strategy is discussed.

3.1 The Super-Resolution Model

As discussed in section 2.3, modern super-resolution techniques are usually based on neural network models. The model I have chosen, developed by Kuleshov et al. in 2017 [5], is not the most the most recent. In fact, more recent models have proven to outperform Kuleshov's model, such as Eskimez's GAN-based solution [6].

However, Kuleshov's slightly outdated model was of all the models I found by far the most accessible, since the source code is publicly available on GitHub[1]. Additional benefits of their model are its simplicity, its end-to-end nature and the fact that it can easily handle any desired sampling rate. The model also supports different upscaling ratio's (2x, 4x and 6x) out of the box. For these reasons Kuleshov's model was preferred over more recent models.

The architecture of Kuleshov's deep neural network highly resembles that of a typical autoencoder [13]. Just like a typical autoencoder network it has a number of convolutional downsampling layers (called D Blocks in figure 2) and an equal number of convolutional upsampling layers (called U Blocks in figure 2). It also has a bottleneck layer, which is typical for autoencoders. The bottleneck layer encourages the model to learn a low dimensional representation of the data, which prevents overfitting on the training data. Unlike typical autoencoders the model has concatenating skip connections between each corresponding down- and upsampling layer as well as additive skip connections between the input and output layers. These skip connections allow the upsampling layers to reuse information embedded in the weights of the downsampling layers.

Where Kuleshov's approach differs most from an autoencoder is in the way it is trained. Autoencoders are trained in an unsupervised manner where the original time series also serves as the ground-truth, while Kuleshov's model is trained with low-resolution audio signals as input and corresponding high-resolution audio signals as output. The low-resolution audio signal is upsampled using interpolation – by means of a 3rd order B-Spline – such that the sample rate of the high-resolution output signal is matched, before it is fed through the network.

A mean squared error (MSE) objective (loss function) is used during training:

$$l(D) = \frac{1}{n}\sqrt{\sum_{i=1}^{n} \|y_i - f_\theta(x_i)\|_2^2} \tag{1}$$

Where D is a dataset consisting of n audio fragments x_i and their corresponding reference fragments y_i. $f_\theta(x_i)$ represents the prediction on input x_i of a model parameterized by θ.

A number of hyper-parameters can be tweaked to change the capacity and (training) behavior of the model:

- **pool size** and **pool stride**, respectively, determine the size and stride of the max-pooling windows used in the convolutional blocks.

[1] https://github.com/kuleshov/audio-super-res

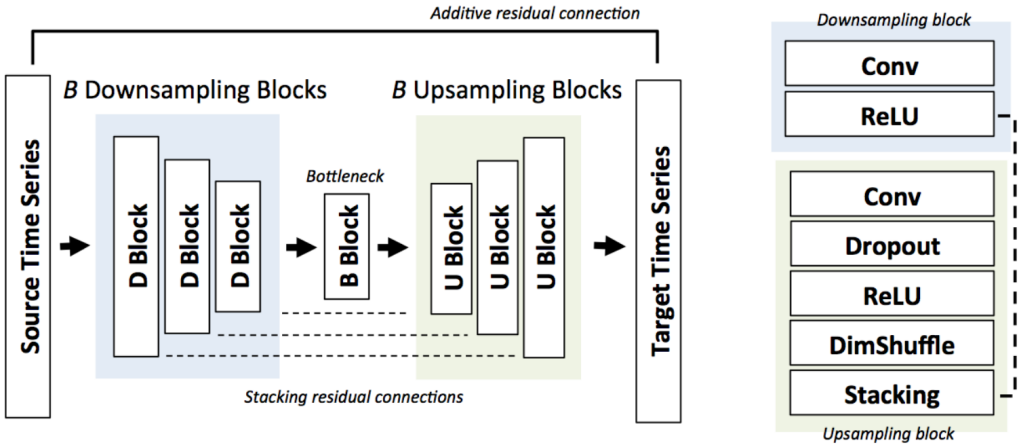

Figure 2: The architecture of the neural network model designed by Kuleshov et al. [5]

- The **layers** parameter (denoted as B in figure 2) determines how many convolutional layers are included in the upsampling and downsampling blocks – e.g. **layers**=4 means there are 4 upsampling and 4 downsampling blocks.
- The **scale** parameter determines the ratio between the low-resolution input signal and the high-resolution output signal.
- **patch size** determines the size of patches in terms of number of samples. These patches are extracted from the audio in training and validation sets during pre-processing (see section 4.1). These patches are then passed through the network one by one to get predicted output per patch.
- **patch stride** determines at which intervals patches are extracted. If **patch stride** is smaller than **patch size**, subsequent patches will have some overlap.
- The **epochs** parameter determines the number of epochs to use during training. During each epoch the entire training set is passed through the network in batches and after each batch the weights of the network are updated based on the prediction error.
- **batch size** determines the size of each batch in numbers of patches. Larger batch sizes use more GPU memory, but speed up the training process.
- The **learning rate** is a scalar that determines by how much the weights are adjusted after each training step. A higher learning rate causes the model to learn more quickly, but allows for less exploration of the problem space – thus a learning rate that is too high might result in sub-optimal model performance.

3.2 The LaRed and ConfLab Datasets

For this research I was provided with two different datasets: the LaRed dataset and the ConfLab dataset. The ConfLab dataset, as mentioned in 2.2, was gathered at a networking event and the audio was sampled at a low frequency (1.25kHz). Since the event where

the data was collected was an international event, we can expect different languages to be contained in the audio. Because of the low sampling rate it is hard to verify what languages are spoken exactly, but most of the spoken audio is probably in English.

The LaRed dataset was gathered at a different networking event, but the audio was recorded at a much higher sampling rate (44.1kHz; CD quality). The LaRed dataset contains audio recorded from the perspectives of 16 different people, totalling roughly 11 hours of speech data (after cutting out silences). Most of the spoken audio is in Dutch, though sometimes English is spoken as well.

Since the LaRed audio is available in very high quality, it is well suited as dataset for training the model. This is because high-resolution audio is needed to generate good reference transcriptions and it allows for testing model on a wide range of input sampling rates – in theory anywhere from 0Hz to $\frac{44.1}{r}$kHz, where r is the upsampling rate. The ConfLab audio, on the other hand, is only available in 1.25kHz sampling rate, which severely limits its usefulness as training dataset.

Therefore, the LaRed data has been used to train and test the model. And a final assessment of the model is made by testing the model on the ConfLab data.

3.3 Evaluation

Naturally, a way is needed to quantify the performance of the model objectively. Widely used metrics for measuring BWE methods in literature are SNR (**S**ignal-to-**N**oise **R**atio), LSD (**L**og **S**pectral **D**istance) and PESQ (**P**erceptual **E**valuation of **S**peech **Q**uality).

SNR usually describes the ratio of a signal to noise in that signal – hence its name. It operates in the time domain, as it divides the squared amplitudes of a reference signal by the squared amplitudes of a noise signal at corresponding discrete time points. Squared amplitudes are proportional to the intensity – power per unit area – of a signal. In this paper SNR is used to compare the intensity of error in predicted audio signals to the intensity of their corresponding reference signals. The following formula is used for SNR in this paper:

$$\text{SNR}(x, y) = 10\log\frac{\|y\|_2^2}{\|x - y\|_2^2} \tag{2}$$

Where y is a reference signal and x is an approximation, thus $\|x - y\|$ can be thought of as the difference, or error, between the approximated signal and the reference signal. Higher SNR values are better. One thing to note is that it has been shown that the SNR measure has very low correlation with subjective tests of speech quality, and is therefore not very indicative of change in intelligibility [14]. The SNR measure is mostly included because it could be useful for comparing my results with results from previous or future papers.

LSD measures the distance between two spectrograms in decibels. This implies that the LSD measure operates in the frequency domain. Instead of comparing amplitudes at discrete time points, LSD compares the power per frequency bin of a reference signal to the corresponding frequency bin of an approximated signal. Before LSD can be measured, first the spectrum of an audio signal must be calculated – using the FFT (**F**ast **F**ourier **T**ransform) method –, this spectrum is then squared and the logarithm is taken of the squared spectrum.

$$\text{LSD} = \frac{1}{L}\sum_{l=1}^{L}\sqrt{\frac{1}{K}\sum_{k=1}^{K}[X(l,k) - \widehat{X}(l,k)]^2} \tag{3}$$

X and $\widehat{X}$ are the log-power spectra (LPS) of the ground truth and estimated signals respectively. K is the number of frequency bins and L is the number of segments in time.

Similarly to SNR, the LSD metric has low correlation with subjective speech quality measures and is not particularly useful for our purposes, but is included for comparison with other papers in the area of audio super-resolution.

PESQ is a measure that was specifically developed for assessing the speech quality in a degraded signal, standardized by the International Telecommunication Union Telecommunication Standardization Sector (ITU-T). PESQ has been shown to correlate significantly better with subjective audio quality measures, as it was in fact designed to do [15]. PESQ emulates the MOS (**M**ean **O**pinion **S**core) metric, which is a measure that is typically used in subjective studies of audio quality. MOS quantifies the subjectively percepted quality of audio using a real-valued scale from 1 to 5, where 1 corresponds to bad audio quality and 5 to excellent audio quality. Because of its high correlation with subjective audio quality measures – which are considered the best way to measure audio quality –, the PESQ metric is more indicative of audio quality and intelligibility than both SNR and LSD [14].

There is no simple formula to calculate PESQ, instead the measurement is performed using proprietary software written in C, for which a Python wrapper is publicly available [16].

While the previously described metrics are somewhat useful for comparing results of one model to the results of another, their use for measuring the intelligibility of generated audio is limited and they are of no use at all for testing whether actual words or sentences can be understood from the generated audio. Thus, to verify whether the model actually increases the ability to understand or transcribe conversations contained in the audio, another way of measuring is needed.

To this end, automatic transcription has been used to generate transcriptions of the audio, which are called hypothesis transcriptions. These hypothesis transcriptions are then compared to reference transcriptions. We can measure how different the hypothesis is from the reference by calculating the word error rate (WER):

$$\text{WER} = \frac{S + I + D}{N} \tag{4}$$

Where S is the number of substitutions (words that are different in the hypothesis transcription), I is the number of insertions (words that are in the hypothesis, but not in the reference), D is the number of deletions (words that are in the reference but not in the hypothesis) and N is the total number of words in the reference transcription.

This WER can be used to compare the intelligibility of audio generated by one trained model to audio generated by another. Lower WER's are better.

4 Experimental Setup and Results

This section describes in detail how the experiments were conducted and concludes with presenting the results that were obtained in the process.

4.1 The Pre-Processing Pipeline

The audio in the LaRed dataset is contained in 16 large audio files (4 hours each), each audio file corresponding to audio of a different participant. First, large silences were cut from the audio, using a custom python script that drops chunks from the audio where the

volume is below a certain threshold – 89dB was found to be a good threshold. This resulted in 16 audio files between 1 and 3 hours in length.

These audio files were then normalized – such that the volume of each audio file is roughly the same –, noise-reduced[2] and cut up into smaller segments of between 0.5 and 100 seconds in length. The normalization, noise-reduction and cutting were all done using Logic Pro X, a proprietary software developed by Apple Inc.[3]. Logic Pro X was chosen because I happened to have access to it and it has a tool for cutting out silences that was significantly faster than solutions using python – a bonus is that it offers a nice user interface which helps when trying to find suitable parameters.

The quality of audio from one of the participants was quite poor as it contained a lot of popping and cracking sounds, so it was left out of the training set. It was, however, used for additional evaluation later on. After all of the previous steps, and excluding the audio from one participant, a total of roughly 10.5 hours of audio remained, consisting purely of speech with reduced background noise.

The dataset was then split up into three parts: a training set, a validation set (for evaluating the model during training) and a test set (for evaluating the model after training). 80% of the files were put into the training set, 10% into the validation set and the remaining 10% into the test set. Files were chosen randomly, without replacement. The test set was used later on to generate all of the results.

Some further processing was needed on the training and validation sets before they could be passed to the model for training. Each audio file was first resampled to the desired output sampling rate (X), these audio files would be used as the high-resolution ground truth samples. Then each of these high-resolution audio files was paired with a low-resolution version with sampling rate $\frac{1}{4} \cdot X$, produced by further decimating the resampled signal by the inverse of the upsampling ratio ($4^{-1} = \frac{1}{4}$) – after applying an order 8 Chebyshev type I filter to avoid aliasing.

Both the low- and high-resolution files were then cut into smaller patches of equal length – determined by the patch size parameter. These patches have some overlap with preceding and succeeding patches – depending on a stride parameter. Some patches were dropped to ensure that the number of patches is a multiple of the batch size – which is required by the model. Finally, the resulting arrays of patches were written to disk, ready for training.

Using this pre-processing pipeline, five training sets were generated, each based on a different input sampling rate: 300Hz, 550Hz, 800Hz, 1250Hz and 2000Hz.

4.2 Training the Model

The neural network was trained on each of the five generated training sets. I have chosen to work with an upsampling ratio of 4, since the creators of the network achieved good results with it. Furthermore, a batch size of 32 (patches) was chosen. The patch size was set to 8192 (samples) and the stride for patching to 4096. I chose to use a learning rate of $3 \cdot 10^{-4}$, pooling window with size 2 and stride 2 and 4 downsampling and upsampling layers. The reason for choosing these values is that the creators of the neural network model [5] achieved good results with them in their experiments and based on exploratory experiments I saw no apparent reason to change them.

[2]Noise reduction was performed using the X-Noise plugin for Logic Pro X, which learns a noise profile based on a user-selected piece of the audio file, containing purely noise, and then applies the noise reduction to the remainder of the audio.

[3]https://www.apple.com/logic-pro/

Input sample rate (Hz)	Number of epochs	Time (hours)
300	60	1.59
550	60	4.56
800	60	10.15
1250	60	18.97
2000	30	20.33

Table 1: Duration of training sessions of the same neural network with different sample rates as input. The training duration is given in hours.

Each model was trained for 60 epochs, since early tests showed that additional epochs did not significantly improve performance of the model. With the exception of the 2000Hz dataset, which was trained for just 30 epochs in the interest of time.

The models were trained on a system running Ubuntu 20.04 with 32GB of RAM and an Nvidia RTX 3080 GPU – with 10 GB of VRAM. Table 1 shows how long each model took to train on this system.

4.3 Results

To evaluate the trained models I used four different metrics, as introduced in section 3.3: SNR, LSD, PESQ and WER. For this evaluation a test set was reserved that contains 1.28 hours of speech data (from 15 different speakers), pre-processed in exactly the same way as the training set. This data was not seen by the model during training, though the model was trained on different data from the same participants.

The results are shown in tables 2 and 3. It can be observed that, as expected, speech quality (PESQ) improves with increasing input sampling rates. This is because simply more frequency information can be represented with higher sampling rates and the model seems to be able to make use of this additional information. And this speech quality improvement seems to also translate to an increase in intelligibility of the predicted high-resolution audio compared to the low-resolution audio, as evidenced by the WER values in table 3. For each input sampling rate there seems to be an improvement int the WER of the predicted audio. Though this improvement is very minimal in the models trained on 300Hz and 550Hz, the 800Hz model already shows a more significant improvement (a 3.3% decrease in WER). The most significant improvements are achieved with the 1250Hz and 2000Hz models, with a 4.3% and a 7.1% decrease in WER respectively.

Though these relative improvements are promising, it must be noted that a WER of 90% or higher is quite poor, especially considering the fact that these WER's are calculated based on generated reference transcriptions – not a true human-made trancription. Thus the true WER would be even worse. This means that only a very small percentage of the actual words are transcribed correctly. I confirmed this by listening to the low-resolution and predicted high-resolution audio files. Most of the audio is hardly comprehensible in the 1250Hz and lower low-resolution audio files, while words and sometimes sentences or parts of sentences start to be comprehensible in the upscaled versions of the 1250Hz and 2000Hz audio files[4].

As briefly mentioned in section 4.1, the audio of one speaker was left out of the training set completely. The audio of this speaker was used to see how well the model performs on

[4]It must be noted that this listening experiment was purely explorational, in future work a more formal subjective study to verify results would be recommended.

Input sample rate (Hz)	SNR (dB)	LSD (dB)	PESQ (MOS)
300	0.27	4.67	1.56
550	1.34	3.64	1.59
800	2.80	3.33	1.73
1250	6.46	3.37	2.33
2000	9.63	3.56	2.67

Table 2: Objective results of the model trained on audio sourced at five different sampling rates from the same dataset. The SNR and LSD metrics were calculated based on the predicted audio sampled at the output rate (4× the input rate), with corresponding ground truth audio segments as reference signals at the same sampling rate. The PESQ measure was calculated by first resampling the audio signals (both the predicted and reference signal) at 8kHz.

Input sample rate (Hz)	WER (LR)*	WER (PR)*
300	0.999	0.993
550	0.999	0.972
800	0.995	0.962
1250	0.972	0.929
2000	0.967	0.896

Table 3: Word error rate results of the model trained on audio sourced at five different sampling rates from the same dataset. WER (LR) and WER (PR) are the word error rates of automated transcription of low-resolution (LR) and predicted high-resolution (PR) audio compared to automated transcription of the same audio but at an 8kHz sampling rate.

unknown data from an *unknown* speaker compared to how it performs on unknown data from *known* speakers. The results of this experiment are shown in tables 4 and 5 and a graph comparing aggregated values from this experiment to aggregated values of the experiment based on data from known speakers is shown in figure 3. It can be seen that in every regard, the models perform slightly worse on data from an unknown speaker, as can be expected. However, some significant improvements in terms of WER are still observed for the 1250Hz and 2000Hz models – a 3.5% and 5% decrease in WER respectively.

Finally, I aimed to perform some evaluation of the model using the ConfLab dataset. This proved to be a challenging task, however, since the ConfLab dataset is only available in a low resolution and to calculate all previously discussed metrics a high resolution signal is needed. This means we cannot use the ConfLab dataset to directly compare how the model

Input sample rate (Hz)	SNR (dB)	LSD (dB)	PESQ (MOS)
300	0.26	5.05	1.30
550	0.55	3.95	1.25
800	1.62	3.33	1.73
1250	4.54	3.50	2.13
2000	8.51	3.59	2.73

Table 4: Objective results of evaluation of the model on data from a speaker that was not in the training distribution. The evaluation was done in exactly the same manner as in Table 2.

Input sample rate (Hz)	WER (LR)*	WER (PR)*
300	1.00	0.989
550	0.999	0.976
800	0.998	0.981
1250	0.978	0.943
2000	0.967	0.917

Table 5: Word error rate results of evaluation of the model on data from a speaker that was not in the training distribution. The values were generated in the same manner as in Table 3.

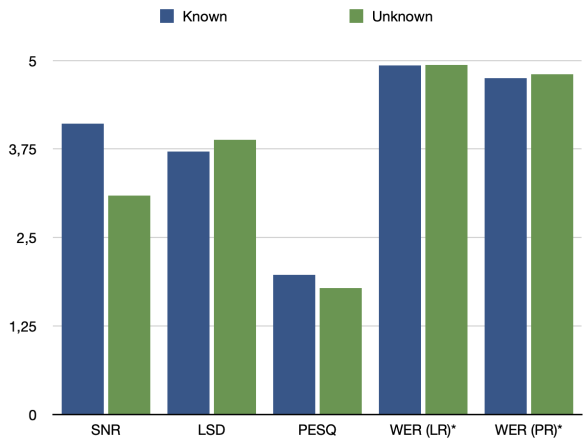

Figure 3: A graph comparing results of evaluation of the model on data from *known* speakers versus data from *unknown* speakers. SNR, LSD and PESQ are all averaged over the different model runs, whereas WER is summed to better accentuate the differences. For SNR and PESQ higher is better, while for LSD and WER lower is better.

performs on out-of-distribution data – e.g. audio recorded at a different time in a different place with a different noise profile.

The best I managed to do was compare spectrograms of the predicted ConfLab audio to spectrograms of the predicted 1.25kHz sample rate version of the LaRed audio visually. An example of such comparison is given in figure 4. A clear difference can be seen in the upper frequency bands – above 600Hz – of the predicted LaRed audio (top right) compared to the predicted ConfLab audio (top left). The LaRed spectrogram displays clear patterns (the stacked areas of brighter yellow spots), while the ConfLab spectrogram looks a lot noisier and no clear patterns are seen. This suggests that the model is having a hard time predicting any useful information in the upper frequency bands. And in fact when comparing the predicted audio files by listening, the LaRed audio is much more intelligible than the ConfLab audio.

This, admittedly quite unscientific, experiment seems to imply that the model does not generalize well on out-of-distribution data. This is no surprise, however, since the model was only trained on data from one dataset, with volume levels normalized and noise reduced in a uniform manner. It can not be expected that the model performs equally well on data from a different dataset with different noise characteristics and speech in different languages.

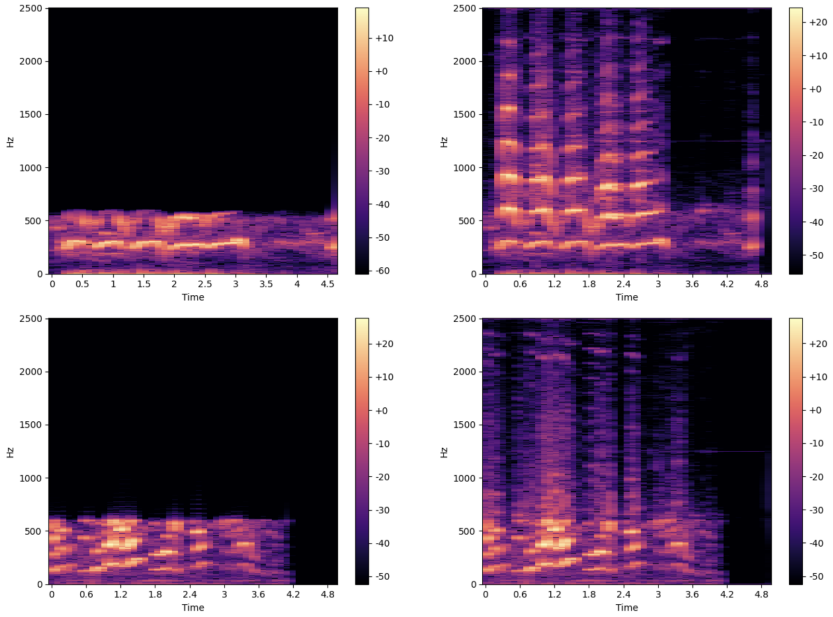

Figure 4: A comparison between audio upsampled from the LaRed dataset (the top two figure) and audio upsampled from the ConfLab dataset. The left two figures are the spectrograms of the low-resolution audio (1.25kHz sample rate). The right two figures are the spectrograms of the predicted high-resolution audio files (5kHz sample rate). Brighter yellow corresponds to higher intensity, whereas darker purple corresponds to lower intensity.

5 Responsible Research

This section reflects on some ethical implications of this research and discusses the reproducibility of the experiments.

5.1 Protecting People's Privacy

As mentioned in the introduction, the topic of privacy is a central part in this research. The aim of recording audio in a low-resolution is to preserve privacy. The goal of this research is to investigate how sound the method of downsampling audio is in terms of preserving privacy, therefore contributing towards a more privacy-centred digitized world.

Since privacy is very important in this research, extra care has been taken when handling the provided datasets. All of the processing has been done on local machines, preventing leakage of the data to third parties.

5.2 Reproducibility of Results

In scientific research it is instrumental that experiments and their results can be reproduced be independent researchers. To this end the process and evaluation strategy have been explained in as much detail as possible, while also taking conciseness into consideration. Additionally, the source code of the super-resolution model, which lies at the core of this research, is publicly available on GitHub.

A difficulty in reproducing results could lie in the fact that the datasets are, as of today, not published. This is a caveat of keeping data private in an effort to preserve privacy of participants. While it might be possible that the datasets will be published one day, until then a good alternative would be to run the experiments on a similar dataset that is publicly available.

Finally, the use of Logic Pro X, has some downsides when it comes to reproducibility. Firstly, Logic Pro X is not free software, and I would not advise anyone to purchase it only for the purpose of reproducing this research. There are, however, very good free alternatives. Such as Audacity[5] or custom python scripts using the *librosa* and *scipy* libraries. A second downside of using Logic Pro X, instead of python scripts, is that there is no easy way to automate and publish the pre-processing process, thus making it harder to reproduce the exact steps taken.

6 Conclusion

The results have shown that a considerable improvement in terms of word error rate (WER) can be achieved using super-resolution on audio sampled at 1.25kHz or 2kHz. The decreased WER can be interpreted as a significant improvement in intelligibility. And with a WER of 92.9% (4.3% lower than the low-resolution baseline) for the predicted 1.25kHz and a WER of 89.6% (7.1% lower than the low-resolution baseline) for the predicted 2kHz audio it is clear that some previously obfuscated words can be revealed from the audio using super-resolution. In an informal experiment it was verified that indeed words start to become more intelligible in the predicted audio – compared to their low-resolution counterparts.

[5] https://www.audacityteam.org

That being said, while individual words and sometimes parts of sentences can be distinguished, with the achieved word error rates it seems like it is not realistic to expect entire conversations or significant parts of conversations to be compromised.

I am not confident in answering the original research question conclusively: *Can existing super-resolution techniques be used to reveal hidden conversations in privacy-sensitive audio?* With the model I used, trained on a limited dataset, I would be inclined to say the answer is *not really*. Since even on data from the same distribution as the training data, the model seems to only be able to reveal very limited parts of conversations. However, I suspect that considerable improvements are possible, using more modern models and training on larger datasets – more discussion on this is provided in section 7.

7 Discussion and Future Work

Some significant limitations have been identified during the research. First of all, as previously mentioned, the model that was used for super-resolution is not the most recent model. More recent models have been proposed that have proven to outperform the model used in this paper [6]–[11]. These models were not publicly available, however, and it would have taken valuable time to implement them from scratch.

Another limitation was the size of the dataset used for training. The VCTK Corpus is a dataset commonly used for super-resolution, containing a total of 44 hours of spoken sentences [17]. This is four times more speech data than in the LaRed dataset.

In addition, the dataset consisted of mostly of Dutch speech. While this is not a problem if the goal is to reveal conversations in Dutch low-resolution speech, it is a serious limitation when the goal is to do the same for audio spoken in a different language. This might very well explain, in part, the poor performance on the out-of-distribution ConfLab dataset.

Finally I would like to note that the research question might have been stated a bit too generally, especially for a pioneering research in the area of using super-resolution to reveal hidden privacy-sensitive conversations. Specifically the part "*hidden conversations*" may have been a bit to fuzzy. After all, when can we safely say that an actual conversation has been compromised, as opposed to just some disjoint words or parts of sentences? It might have been more apt to limit this research to investigating the possibility of revealing single words, which would have been much easier to verify.

7.1 Recommendations

Based on the identified limitations, I would like to make some recommendations for future research. I believe significant performance improvements can be made by improving on three aspects: the model, the dataset and the training procedure.

An obvious low-hanging fruit is to adopt a more recent super-resolution model. It takes some effort to make a custom implementation, but might be well worth the effort. An alternative might be to ask the creators of an existing solution for access to their super-resolution model.

Another improvement would be to use a larger dataset and preferably even multiple datasets gathered under different circumstances. Using multiple datasets or by preprocessing parts of the same dataset in different ways a model could be achieved that is able to generalize better than the model in this paper. Of course the capacity of the model needs to be adjusted as well, to accommodate the added complexity. Additionally, if there

is a desire to operate the model on multiple languages, then of course it must be trained on multiple languages as well.

Finally, a truly ground-breaking approach would be to somehow train a super-resolution model in conjunction with an ASR (**A**utomated **S**peech **R**ecognition) model using a shared objective. This might allow the super-resolution model to optimize on improving intelligibility more directly during training, reinforcing its ability to reveal actual conversational information.

References

[1] University of Technology Sydney, AU, M. J. Anwar, A. Q. Gill, University of Technology Sydney, AU, G. Beydoun, and University of Technology Sydney, AU, "A review of information privacy laws and standards for secure digital ecosystems", in *Australasian Conference on Information Systems 2018*, University of Technology, Sydney, 2018. DOI: 10.5130/acis2018.bb. [Online]. Available: https://utsepress.lib.uts.edu.au/site/chapters/10.5130/acis2018.bb/ (visited on 06/06/2022).

[2] O. Lederman, A. Mohan, D. Calacci, and A. S. Pentland, "Rhythm: A unified measurement platform for human organizations", *IEEE MultiMedia*, vol. 25, no. 1, pp. 26–38, Jan. 2018, Conference Name: IEEE MultiMedia, ISSN: 1941-0166. DOI: 10.1109/MMUL.2018.112135958.

[3] "ConfLab - ACM MM 2019". (2019), [Online]. Available: https://conflab.ewi.tudelft.nl/ (visited on 04/26/2022).

[4] N. R. French and J. C. Steinberg, "Factors governing the intelligibility of speech sounds", p. 31,

[5] V. Kuleshov, S. Z. Enam, and S. Ermon, "Audio super resolution using neural networks", *arXiv:1708.00853 [cs]*, Aug. 2, 2017. arXiv: 1708.00853. [Online]. Available: http://arxiv.org/abs/1708.00853 (visited on 04/26/2022).

[6] S. E. Eskimez, K. Koishida, and Z. Duan, "Adversarial training for speech super-resolution", *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 347–358, May 2019, ISSN: 1932-4553, 1941-0484. DOI: 10.1109/JSTSP.2019.2909077. [Online]. Available: https://ieeexplore.ieee.org/document/8681126/ (visited on 06/05/2022).

[7] G.-X. Lin, S.-W. Hu, Y.-J. Lu, Y. Tsao, and C.-S. Lu, "QISTA-Net-Audio: Audio Super-Resolution via Non-Convex ℓ_q-Norm Minimization", in *Proc. Interspeech 2021*, 2021, pp. 1639–1643. DOI: 10.21437/Interspeech.2021-670.

[8] W. J. Jose, "AMRConvNet: AMR-coded speech enhancement using convolutional neural networks", in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Toronto, ON, Canada: IEEE, Oct. 11, 2020, pp. 1671–1676, ISBN: 978-1-72818-526-2. DOI: 10.1109/SMC42975.2020.9283346. [Online]. Available: https://ieeexplore.ieee.org/document/9283346/ (visited on 05/07/2022).

[9] Y. Li, M. Tagliasacchi, O. Rybakov, V. Ungureanu, and D. Roblek, "Real-time speech frequency bandwidth extension", in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada: IEEE, Jun. 6, 2021, pp. 691–695, ISBN: 978-1-72817-605-5. DOI: 10.1109/ICASSP39728.2021.9413439. [Online]. Available: https://ieeexplore.ieee.org/document/9413439/ (visited on 05/07/2022).

[10] C.-D. Xu, X.-P. Ling, and D.-W. Ying, "Codec network for speech bandwidth extension", in *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, Nanchang, China: IEEE, Mar. 26, 2021, pp. 387–391, ISBN: 978-1-66541-540-8. DOI: 10.1109/ICBAIE52039.2021.9389968. [Online]. Available: https://ieeexplore.ieee.org/document/9389968/ (visited on 05/07/2022).

[11] J. Su, Y. Wang, A. Finkelstein, and Z. Jin, "Bandwidth extension is all you need", in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada: IEEE, Jun. 6, 2021, pp. 696–700, ISBN: 978-1-72817-605-5. DOI: 10.1109/ICASSP39728.2021.9413575. [Online]. Available: https://ieeexplore.ieee.org/document/9413575/ (visited on 05/07/2022).

[12] National Institute of Technology Warangal, Warangal-506004, India, N. Prasad, and T. K. Kumar, "Bandwidth extension of speech signals: A comprehensive review", *International Journal of Intelligent Systems and Applications*, vol. 8, no. 2, pp. 45–52, Feb. 8, 2016, ISSN: 2074904X, 20749058. DOI: 10.5815/ijisa.2016.02.06. [Online]. Available: http://www.mecs-press.org/ijisa/ijisa-v8-n2/v8n2-6.html (visited on 06/08/2022).

[13] D. Bank, N. Koenigstein, and R. Giryes, *Autoencoders*, Number: arXiv:2003.05991, Apr. 3, 2021. arXiv: 2003.05991[cs,stat]. [Online]. Available: http://arxiv.org/abs/2003.05991 (visited on 06/12/2022).

[14] A. Rix, "Perceptual speech quality assessment - a review", in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, Montreal, Que., Canada: IEEE, 2004, pp. iii–1056–9, ISBN: 978-0-7803-8484-2. DOI: 10.1109/ICASSP.2004.1326730. [Online]. Available: http://ieeexplore.ieee.org/document/1326730/ (visited on 06/17/2022).

[15] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs", in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 2, Salt Lake City, UT, USA: IEEE, 2001, pp. 749–752, ISBN: 978-0-7803-7041-8. DOI: 10.1109/ICASSP.2001.941023. [Online]. Available: http://ieeexplore.ieee.org/document/941023/ (visited on 06/17/2022).

[16] M. Wang, C. Boeddeker, R. G. Dantas, and ananda seelan, *ludlows/python-pesq: supporting for multiprocessing features*, version v0.0.4, May 2022. DOI: 10.5281/zenodo.6549559. [Online]. Available: https://doi.org/10.5281/zenodo.6549559.

[17] (:Unkn) Unknown, *Superseded - cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit*, en, 2017. DOI: 10.7488/DS/1994. [Online]. Available: https://datashare.ed.ac.uk/handle/10283/2651.