Master thesis on Sound and Music Computing

Universitat Pompeu Fabra

Deep Noise Suppression for Real Time Speech Enhancement in a Single Channel Wide Band Scenario

Esteban Gómez

Supervisor: Andrés Pérez

Co-Supervisor: Pritish Chandna

September 2021

Universitat Pompeu Fabra Barcelona

Master thesis on Sound and Music Computing

Universitat Pompeu Fabra

Deep Noise Suppression for Real Time Speech Enhancement in a Single Channel Wide Band Scenario

Esteban Gómez

Supervisor: Andrés Pérez

Co-Supervisor: Pritish Chandna

September 2021

Contents

Dedication

Dedicated to my Chilean family for giving me the possibility of flying away (literally) and to my Spanish family for making me feel at home ever since.

Acknowledgement

I would like to express my deepest gratitude to my supervisor Andrés Pérez and co-supervisor Pritish Chandna for providing feedback, help and expertise on the topics of speech enhancement and deep learning during the course of this project, especially considering the ever-changing scenario caused by the ongoing pandemic. In addition, I would like to thank Voicemod for their invaluable support providing all the necessary hardware to carry out this research, without which this thesis would have not been possible. Inestimable thanks are also due to the staff, researchers and faculty of the Music Technology Group at Universitat Pompeu Fabra, for their commitment regardless of the unusual academic circumstances. Last but not least, I would like to thank Microsoft Corporation, for allowing me to use their proprietary tools in this research.

Abstract

Speech enhancement can be regarded as a dual task that addresses two important issues of degraded speech: Speech *quality* and speech *intelligibility*. Improved speech *quality* can reduce listener's fatigue, whereas improved speech *intelligibility* can reduce the listener's effort to understand and extract meaning from speech. This work is focused on speech *quality* in a real time context. Algorithms that improve speech quality are sometimes referred to as *noise suppression algorithms*, since they enhance quality by suppressing the background noise of the degraded speech. Improving state of the art noise suppression algorithms could lead to significant benefits to several applications such as video conferencing systems, phone calls or speech recognition systems. Real time capable algorithms are especially important for devices with a limited processing power and physical constraints that cannot make use of large architectures, such as hearing aids or wearables. This work uses a deep learning based approach to expand on two previously proposed architectures in the context of the Deep Noise Suppression Challenge carried out by Microsoft. This challenge has provided datasets and resources to teams of researchers with the common goal of fostering the research on the aforementioned topic. The outcome of this thesis can be divided into three main contributions: First, an extended comparison between six variants of the two selected models, considering performance, computational complexity and real time efficiency analyses. Secondly, making available an open source implementation of one of the proposed architectures as well as a framework translation of an existing implementation. Finally, proposed variants that outperform the previously defined models in terms of denoising performance, complexity and real time efficiency.

Keywords: Speech enhancement; Speech quality; Noise suppression; Deep learning; Real time applications

Chapter 1

Introduction

Speech enhancement (SE) can be defined as a task that addresses two important issues with degraded speech: Improve speech *quality* and improve speech *intelligibility*. Higher speech *quality* can reduce listener's fatigue [1] and higher speech *intelligibility* can reduce the listener's effort to understand and extract meaning from speech [2]. Even when these tasks may intuitively seem highly correlated, improving one of them does not necessarily have a positive effect on the remaining one and therefore, they are addressed as two separate tasks. This work is focused on speech *quality* in a real time context. Speech enhancement algorithms that deal with this task are sometimes referred to as *noise suppression algorithms* since they enhance quality by suppressing noise of the degraded speech.

The solution to the general problem of speech enhancement is highly dependent on the application, the characteristics of the noise or source of interference and its relationship with the clean signal. Noise classes can range from steady noises where a pattern can be identified within a short period of time (e.g. fan noise) to convolutive interference (e.g. reverb) or competing sources (e.g. different speakers talking simultaneously). Figure 1 shows the spectrogram of a clean speech signal and different types of degradation.

With the increase of processing power, deep learning methods for speech enhancement have demonstrated auspicious results, outperforming previously existing meth-

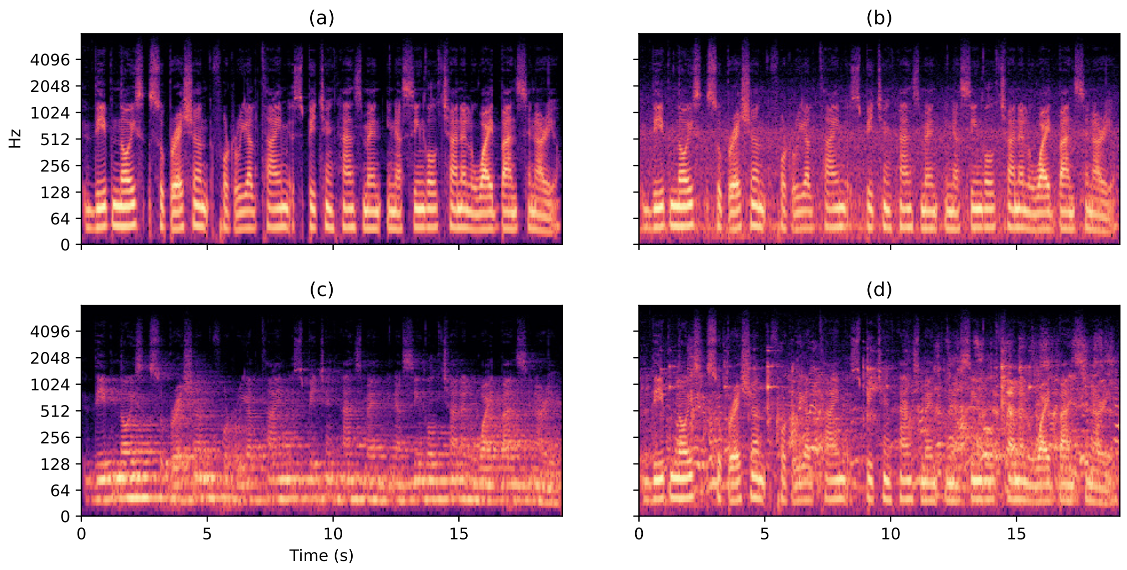

Figure 1: Spectrogram of clean speech (a) and degradation caused by different noise classes. Clean speech and fan noise (b), reverberant clean speech (c) and clean speech combined with additional subject's speech (d). The x axis represents time in seconds and the y axis represents the frequency in hertz.

ods such as spectral subtractive algorithms [3, 4] or statistical model-based algorithms [5, 6]. Deep learning methods are usually based on recurrent neural networks (RNNs) [7, 8, 9] and convolutional neural networks (CNNs) [10, 11, 12, 13], and are believed to be effective in terms of noise reduction due to their effective capabilities for temporal modeling. However, these models often result in large architectures that are not suitable for real time signal processing on a consumer laptop where the processing power has to be shared among several different tasks or on edge devices such as hearing aids or wearables. This leads to the following question: *What deep learning model could provide the best performance under a given set of computational specifications?*

In order to answer this research question, further concerns should be addressed, such as defining the limitations to take into account for a suitable real time deployment, as well as gathering the most adequate data to train different deep learning models. In this regard, the strategies and assets utilized in this thesis expand on the proposed framework given by the Deep Noise Suppression Challenge that is being carried out by Microsoft [14]. This challenge has greatly fostered the progress in the field of speech enhancement by providing resources for research teams to work on the development and testing of new algorithms.

1.1 Motivation

People are faced on a daily basis with devices and applications that currently make use of some form of speech enhancement or can be improved by having an efficient speech enhancement algorithm available. Phone calls, speech recognition systems or video conferencing applications have become increasingly important throughout the years. Due to the COVID-19 pandemic, a significant amount of the worldwide workforce had to rapidly transition and adapt to work remotely. A similar situation happened in the academic environment where classes quickly adopted an online format. This caused an abrupt peak in the use of video conferencing systems. As an example, the popular video conferencing application Zoom, reported a usage of 3.3 trillion meeting minutes during Q3 fiscal year 2021 quarter to date, compared to 79 billion during Q2 fiscal year 2020, and 6 billion during June 2016 [15]. Additionally, such precedent may be the precursor toward an upcoming trend in the work environment: According to Owl Labs, 50% of people will not return to jobs that do not offer remote work options post-COVID-19 [16]. Moreover, 76% of people say they avoid the office entirely (if possible) when they have something important to finish, according to Atlassian [17]. Finally, 80% of workers also say that remote work has increased their work satisfaction [18].

Besides the increase in the usage of video conferencing software, devices such as hearing aids or wearables can also get benefits from efficient speech enhancement algorithms suitable for operating in real time given its specific computational resources and requirements. Deep learning methods applied to speech enhancement are still considered to be in a nascent stage, because there is significant room for improvement in terms of perceptually motivated metrics such as Mean Opinion Score (MOS) in order to achieve a result that is considered ideal [19]. The potential impact that such results may bring into the various aforementioned aspects of our daily life, highlights the relevance of this topic.

1.2 Structure of this thesis

The content of the next sections of this thesis project are organized as follows: **Chapter 2** presents information about the state of the art of speech enhancement. Existing knowledge driven - prior to data driven methods - are briefly described. Deep learning based methods, existing datasets and evaluation methods are the main focus of this chapter. **Chapter 3** describes the methodology used throughout this thesis. It addresses the deep learning model selection, datasets utilized to train, validate and test the models as well as the evaluation procedures carried out to characterise each model. **Chapter 4** contains the results obtained after performing all the steps described in the previous chapter. Information about the performance, real time efficiency and complexity of each individual model is shown and described. Additionally, a comparative evaluation among all models is presented. Then, **Chapter 5** discusses the results obtained in Chapter 4, addressing the outcome and contributions of this thesis as well as the prospects of a hypothetical production setting where these models may be used. Potential directions of future work are also discussed. Finally, conclusions and final remarks of this research are summarized. An accompanying online repository containing the code used in this thesis is also available[1].

[1]https://github.com/eagomez2/upf-smc-speech-enhancement-thesis

Chapter 2

State of the art

Potential solutions for speech enhancement are largely dependant on the application, the characteristics of the noises or sources of interference and the relationship between these and the clean signal. Noises can be of various types: in some cases (e.g. fan noise) a pattern can be identified within a short period of time, whereas in other cases the interference is of convolutive nature (e.g. reverb) or directly competes with the target signal (e.g. different speakers talking simultaneously) and therefore the appropriateness of a given solution will depend on these factors.

A significant number of different approaches to speech enhancement have been proposed throughout the years. Weiss et al. (1974) [3] and Boll (1979) [4] proposed *spectral subtractive algorithms* where the main assumption is that the noise is additive and therefore by determining the noise spectrum without speech, it can be subtracted from the mixture. Later on, *statistical-model-based algorithms* were proposed by McAulay et al. (1980) [5] and Ephraim et al. (1984) [6]. These algorithms exploit the idea that given a set of measurements of the Fourier transform coefficients of a signal, it is possible to find a linear or nonlinear estimator of the transform coefficients of the clean signal. Later on, Weintraub (1985) [20] and almost a decade later, Brown et al. (1994) [21] proposed *binary mask algorithms* that consist of finding a binary gain function that can help to discard the Fourier transform bins that do not belong to the speech signal.

Currently and due to the increase of computing power, speech enhancement research has moved towards data driven approaches based on deep learning. These methods have demonstrated a great potential and outperformed the previously existent techniques. In 2014, Xu et al. presented a regression based speech enhancement framework using deep neural networks (DNNs) [22]. Other early deep learning methods included recurrent neural networks (RNNs) that were believed to be promising in terms of efficiency due to their effective capabilities of temporal modeling [9]. While such models seemed to hit a performance saturation, convolutional recurrent neural networks (CRNs) and convolutional neural networks (CNNs) turned out to be a good alternative, but resulted in very large model architectures, rendering them impractical for edge devices such as consumer laptops, mobile phones or wearables and hearing aids [23]. Due to the wide variety of computing power present in different consumer devices, it is specially important to find an adequate speech enhancement algorithm for each case, given its specific processing speed limitations and data availability. A selection of the aforementioned methods will be summarized in the following sections.

2.1 Spectral-subtractive algorithms

These algorithms are based on the assumption that the noise is additive and therefore an estimate of the clean signal spectrum can be obtained by subtracting the estimate of the noise spectrum from the noisy speech signal. When the speech signal is absent, the noise estimate can be updated. Additionally, these algorithms assume that noise is stationary or slowly varying and therefore no abrupt spectral changes are expected during adjacent frames of audio.

2.2 Statistical-model-based algorithms

These algorithms are focused on the nonlinear estimation of the magnitude spectrum of the clean speech signal using different statistical models and optimization criteria. Given a set of measurements that depend on an unknown parameter, the goal is to find a nonlinear estimator of the target parameter of interest. In speech enhancement, the measurements correspond to the STFT coefficients of the noisy spectrum and the parameters of interest are the STFT coefficients of the clean speech signal. The most popular technique to achieve this is to obtain the maximum-likelihood (ML) estimators [24].

2.3 Binary mask algorithms

Binary mask algorithms make use of binary gain functions, which corresponds to selecting a subset of frequency bins (sometimes referred to as *channels*) from the magnitude spectrum of the corrupted speech and discarding the rest. A prescribed rule or criterion - that is not unique - is used for the selection of bins. As opposed to previous algorithms, when applied to corrupted speech spectra, this algorithm can produce large gains in speech intelligibility. These algorithms have been originated in the field of computational auditory scene analysis [21, 20] and have been used also in automatic speaker recognition (ASR) [25] rather than directly being born for noise suppression. The first demonstration of improvements in terms of speech intelligibility in monoaural conditions was shown in [26].

A binary mask is essentially a matrix with the same size as the spectrogram that contains gain values of 0 and 1. A value of 1 will allow the sound from the mixture through and a value of 0 will discard a specific bin. The binary mask is then applied to the mixture as an element-wise multiplication. This method makes the assumption that any bin is dominated either by speech or by noise. This assumption is also known as *W-disjoint orthogonality* [27]. Other types of masks such as soft masks can be also found in source separation problems and could be applied to speech enhancement as well [28].

2.4 Deep learning methods

With the increase of available computing power, deep learning based methods for speech enhancement became increasingly popular because of their promising results. Additionally, speech enhancement can be assumed as a particular case of source separation and therefore, solutions given to both tasks are jointly moving forward. Furthermore, advances from image processing have served as inspiration to novel architectures as in the case of PixelCNN [29] after which WaveNet [30] was conceived, and later on used for both speech denoising [31] and source separation [32]. In the following subsection, a selected number of state of the art architectures and datasets are presented and described.

2.4.1 NSNet2 (Recurrent Neural Network)

NSNet2 was initially proposed as the baseline architecture for the Microsoft DNS-Challenge 2021 [14]. It is a recurrent neural network (RNN) architecture based on gated recurrent units (GRUs) and feed forward (FF) layers. The input is the logarithmic power spectrum that then is normalized. The STFT size used is fixed to 512 with 32ms square-root Hann windows and 16ms of hop size. 0th and Nyquist bins are discarded. After the first FF embedding layer and two GRUs, there are three FF *mapping layers* that use ReLU activation, except for the last one that uses sigmoid activation to ensure a positive output to estimate the suppression gain. This architecture has 2.8M parameters and is shown in Figure 2. This model uses a dynamically compressed loss as proposed in [33]. A compression exponent between 0 and 1 is applied to the magnitude spectra and these are combined with the phase factors. Additionally, a magnitude only loss is mixed with the complex compressed loss based on the value of α as expressed in (2.1) producing a loss function known as *Complex Compressed Mean Square Error Loss.*

$$\mathcal{L} = \alpha \sum_{k,n} ||S|^c e^{j\phi_S} - |\hat{S}|^c e^{j\phi_{\hat{S}}}|^2 + (1-\alpha) \sum_{k,n} ||S|^c - |\hat{S}|^c|^2 \tag{2.1}$$

S corresponds to the original signal and $\hat{S}$ is the reconstructed signal. $\alpha = 0.3$ and $c = 0.3$ are typical values found in literature. A common issue for signal-dependent loss functions is the dependency on the level of the signals which may impact the loss over batch with different dynamic range differences. To overcome this, the signals can be normalized by the active signal level of each utterance before computing the

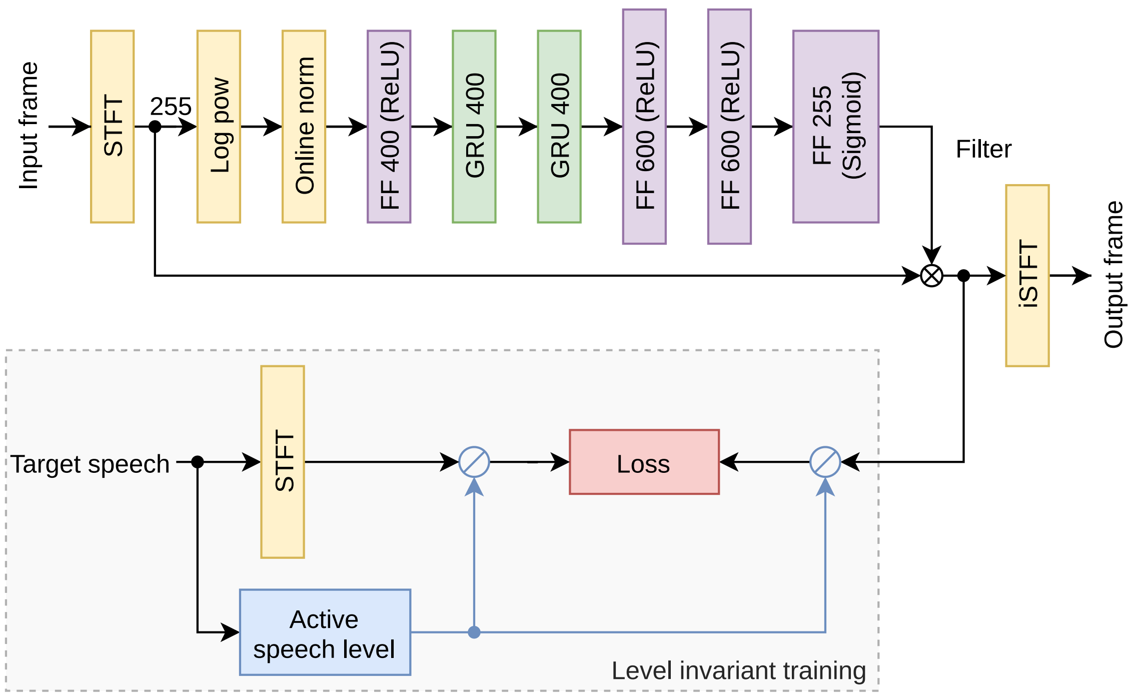

Figure 2: NSNet2 diagram. The gray dotted square encloses the section of the network used to compute the training loss.

loss (in this case the target speech signal standard deviation is computed only for active speech frames).

2.4.2 DTLN (Dual-signal Transformation LSTM network)

As introduced in [34], this architecture is constituted by two *separation cores*, each containing two long short-term memory (LSTM) layers followed by a fully connected layers (FC or also found as FCL in literature) and a sigmoid activation to create a soft mask output. Before the first core, the STFT is computed, and at the end of the core, it is used for synthesis. The frames coming from the first core of the network are then processed by a 1D convolutional layer (1D-Conv) to create a feature representation that is processed by an instant normalization layer (previously introduced as channel-wise normalization in [35]) before being fed to the second separation core. Finally, the result is used as input to an additional 1D-Conv to get the estimate time domain signal that then is reconstructed through an overlap add procedure. The number of parameters of the originally proposed variants range from 984K to 987K. This architecture is shown in Figure 3.

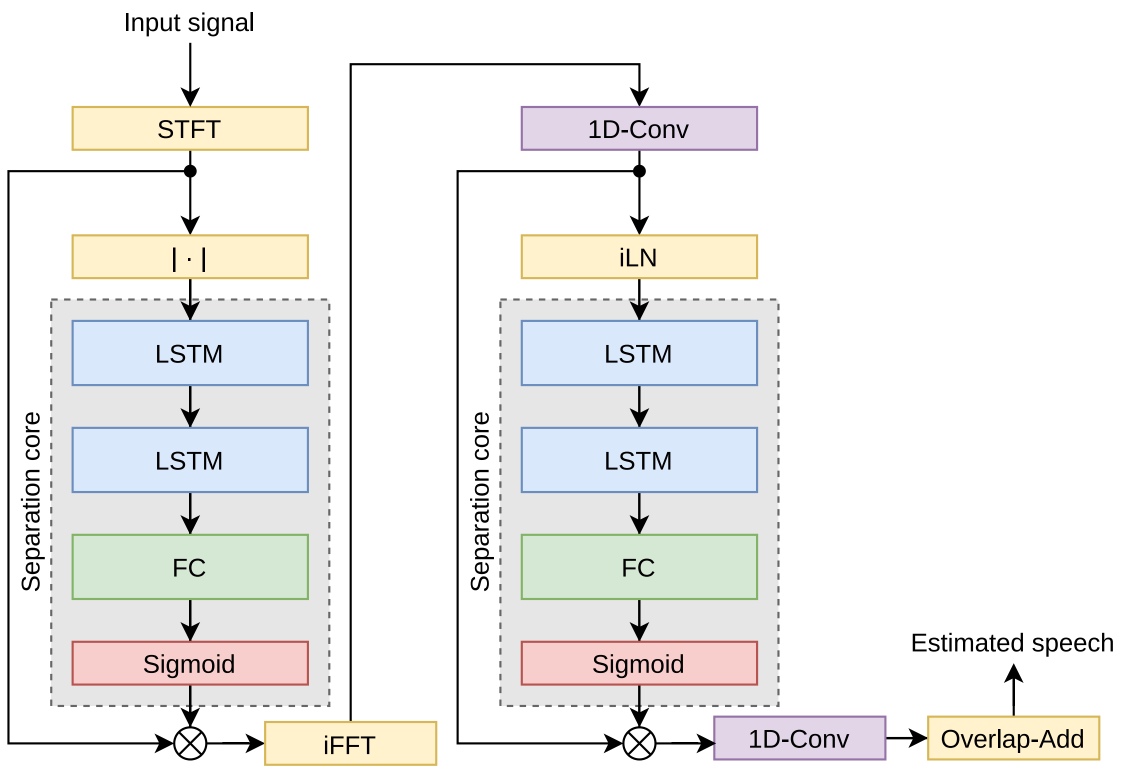

Figure 3: Dual-signal Transformation LSTM Network (DTLN). Each gray block corresponds to a separation core made of two stacked LSTM and a fully connected layer with a sigmoid activation function.

This model utilizes the *Scale-sensitive Negative Signal-to-noise Loss*:

$$f(y,\hat{y}) = -10log_{10}\left(\frac{\sum_t y_t^2}{\sum_t (y_t - \hat{y}_t)^2}\right) \tag{2.2}$$

Where y is the original signal and $\hat{y}$ the reconstructed signal.

Compared to negative SI-SDR used to train other networks such as ConvTasNet [35], the negative SNR loss has the advantage that the scale of separated sources is preserved and consistent with the mixture.

2.4.3 CRUSE (Convolutional Recurrent U-net for Speech Enhancement)

CRUSE is derived from [11] and is designed to leverage both the efficient temporal modeling capabilities of RNNs and the performant yet large architectures of speech enhancement CNN models. This architecture has L symmetric convolutional and transposed convolutional layers (encoder and decoder, respectively) with kernel size (2, 3) in time and frequency dimensions [23]. The convolution kernels have a stride of (1, 2) contributing to downsample the features of the frequency axis. The number of channels C_l (with $l = 1, ..., L$) increase per encoder layer and decrease per decoder layer. Each convolutional layer uses a leaky ReLU activation and the last layer uses a sigmoid activation function. A recurrent layer is situated in the bottleneck between encoder and decoder. The inputs of this layer are all flattened features along the channels. A single GRU has demonstrated significant computational savings as opposed to LSTM or other types of layer. The performance of CRUSE scales with its bottleneck size, however, the complexity of RNN layer scales with R^2, therefore, it implies a trade off between speed and computational requirements. Additionally, parallel disconnected GRUs can be used in the bottleneck as suggested in [36]. In this case, the encoder output is reshaped to be equally distributed among all the GRUs and then is reshaped again to the number of channels in the decoder. This configuration presents the advantage of potential parallel execution. Finally, it has been found that performance improvements can be observed by adding skip connections. Additionally, a trainable channel-wise scaling and bias can be implemented as a convolution with a (1,1) kernel, connecting each encoder layer output to its corresponding decoder layer input. The loss function used for this model is shared with NSNet2 and shown in equation (2.1).

2.5 Datasets

As mentioned at the beginning of the current chapter, noises can be of several types and origins. Having the separate constituents of a noisy speech scenario (i.e. the speaker's recording, the noise recording and the impulse response of a room) it is possible to generate a good estimate of how a similar situation of natural origin would sound. This makes speech enhancement and appropriate candidate for data augmentation. This technique could be understood as a method used to increase the amount of data by combining or modifying copies of existing data or previ-

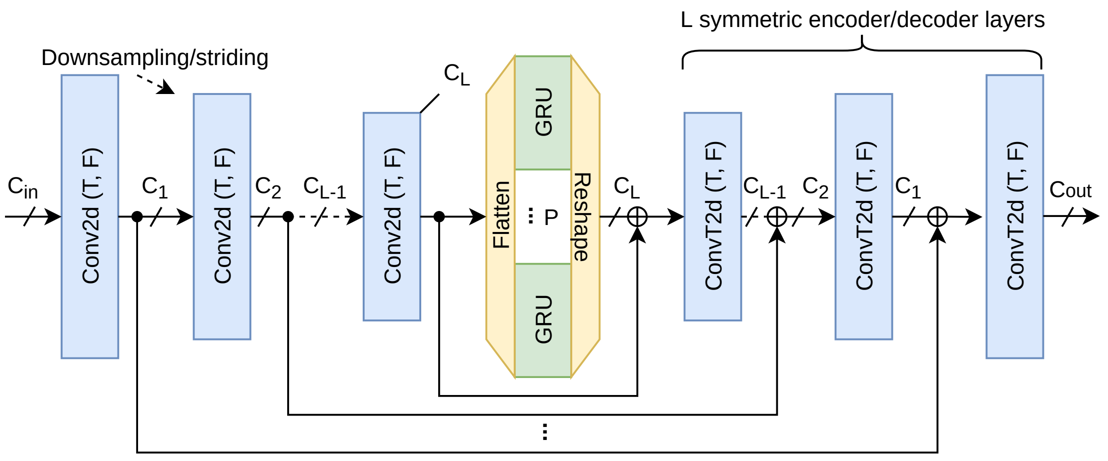

Figure 4: Convolutional Recurrent U-net for Speech Enhancement (CRUSE) architecture with L encoder and decoder layer. P represents the parallel recurrent layers in the bottleneck.

ously generated synthetic data. It is widely used in speech enhancement due to its demonstrated benefits [37]. For this reason, datasets are commonly accompanied with scripts to facilitate the generation of clean/noisy speech pairs under different constrains in terms of parameters such as signal-to-noise ratio (SNR) or reverberation time (RT60) in the case of reverberant spaces.

2.5.1 DNS-Challenge dataset

To the best of the author's knowledge, 2021 Deep Noise Suppression Challenge organized by Microsoft [14] provides one of the most comprehensive collections of publicly available datasets for speech enhancement. It contains neutral clean speech, singing clean speech as well as emotional and non-English clean speech. It also contains samples of noises, and real and synthetic impulse responses. All the samples for the wide band track are recorded or downsampled to 16kHz. For the clean speech, it contains 10000 public domain audiobooks in various languages (mainly English) of 11350 speakers from the Librivox dataset [38]. For the this dataset, 10 audio segments of each book (10 seconds each) were sampled. For singing clean speech, it contains voices recorded in noise-free conditions by 9 male and 11 female professional singers. It contains recordings of vowels, scales, arpeggios, long tones, excerpts and extended techniques. In the case of emotional speech, it is derived from the Crowd-sourced Emotional Multimodal Actors Dataset (CREMA-D) [39] that contains 7442 audio clips from 48 male and 32 female actors between 20 and 74 years old from diverse cultural backgrounds and comprises 3.5 hours of audio. Actors read from a pool of 12 sentences with text accounting for 6 emotions (anger, disgust, fear, happy, neutral and sad) with 4 intensity levels (low, medium, high and unspecified). This dataset was annotated by 2443 human raters. Non-English clean speech contains tonal and non-tonal languages including Chinese (Mandarin), German and Spanish, with over 220 hours total and over 400 native speakers. Regarding noises, the dataset contains 60000 clips of 150 audio classes and was augmented with additional 10000 noise clips downloaded from FreeSound [40] and DEMAND [41] databases, providing 181 hours of noise data in total. Additionally, the dataset contains 3076 recorded impulse responses and approximately 115000 synthetic room impulse responses from openSLR25 and openSLR28 datasets [42].

2.5.2 WHAM! and WHAMR! datasets

The WSJ0 Hipster Ambient Mixtures (WHAM!) dataset pairs each two-speaker mixture in the wsj0-2mix [43] dataset with a unique noise background scene. The mixtures are created by applying a random noise to achieve differences between 0 and 5dB between speech signals prior to the mix. The dataset contains 20,000 two-speaker mixtures for training (30 hours), 5,000 for validation (10 hours) and 3,000 for test (5 hours). There are four variations of wsj0-2 mix dataset, a *min version* where the longer of the two speaker signals is truncated, a *max version* where silence is appended to the shorter signal, and both are available at 16kHz and 8kHz. There is also an additional three-speaker version of wj0-mix. The background noise data was recorded in urban environments such as coffee shops, restaurants, bars, office buildings and parks in the San Francisco Bay Area. The original audio of these environments was recorded at 48kHz and later on downsampled to 16kHz and 8kHz.

WHAMR! extends WHAM! by introducing reverberation to the speech sources. The utilized room impulse responses were generated using the *pyroomacoustics* [44] package. The chosen reverberation times aim to approximate domestic and classroom environments and places with similar reverb to restaurants and coffees where the noises from the WHAM! dataset were recorded.

2.5.3 Other datasets

In addition to the previously mentioned resources, there are other datasets that contain recordings of clean speech that can be used along with data augmentation scripts to produce the noisy speech to be used as target and source, respectively. An example of this case is the The Flickr Audio Caption Corpus contains 40000 spoken captions of 8000 natural images that were collected in 2015 to investigate multimodal learning schemes for unsupervised speech pattern discovery [45]. Another example is the Telecommunications and Signal Processing Laboratory from McGill University that provides the TSP speech that contains over 1400 utterances spoken by 24 speakers (half of them male, half female) that were recorded in an anechoic room and are distributed in several samples rates, including 16kHz [46].

2.6 Evaluation methods

Subjective listening tests provide perhaps the most reliable method for assessing speech quality or speech intelligibility. These tests, however, can be time-consuming and require access to trained listeners [1]. Ideally, the objective measures should be able to assess the quality (or intelligibility) of the enhanced speech without needing access to the original speech signal (known as non-intrusive metrics). In practice, many utilized metrics are intrusive and it is necessary to have access to both the clean and predicted speech. Since the present work is focused on speech quality, the evaluation methods described in the following sections will also be mainly concentrated in this aspect of speech signals. It is important to mention that the separation between objective and subjective quality measures has become increasingly blurry. Some state of the art metrics described below utilize different techniques (sometimes based on deep neural networks) to mimic a trained listener. Even when these are considered objective measures of quality, they are based on data or pursue a high correlation with Mean Opinion Score (MOS) ratings by human listeners [47].

2.6.1 MOS (Mean Opinion Score)

Mean Opinion Score (MOS) is a subjective measure used typically in - but not limited to - telecommunications engineering and represents an overall quality of a system. In the case of noise suppressors, a trained listener would assess the resulting quality of a denoised recording. ITU-T P.800.1 [47] defines different uses of this score. It is a single integer number in the range of 1 to 5, where the lowest quality is 1 and the highest perceived quality is 5. If various listeners have evaluated a single algorithm, the final MOS would be the arithmetic mean over all the evaluation scores and could therefore produce a non-integer value when averaged. Despite of being a subjective measure and given its effectiveness in evaluating noise suppressors, some of the metrics defined in the following subsections attempt to mimic this procedure or to be mapped to a MOS. Since subjective metrics involve a significant number of people listening a usually large amount of samples in order to perform the evaluation, it is deemed as a tedious and expensive task and therefore replicating these mechanisms contributes to alleviate the costs of MOS tests and helps to have a better automated estimate of the performance of different methods.

2.6.2 Segmental signal-to-noise ratio measures

Segmental signal-to-noise-ratio SNR_{seg} can be evaluated in time or frequency domain. It requires the original and processed speech signals to be aligned in time and that phase errors are corrected. In time domain, the initial definition was improved by Richards [48] by shifting the original log function by 1 to avoid large values during silences in the recordings. This improved formula can be usually found in literature as SNR_{seg_R} and is shown in (2.3).

$$SNR_{seg_R} = \frac{10}{M} \sum_{m=0}^{M-1} log_{10} \left(1 + \frac{\sum\limits_{n=Nm}^{Nm+N-1} x^2(n)}{\sum\limits_{n=Nm}^{Nm+N-1} (x(n) - \hat{x}(n))^2} \right) \tag{2.3}$$

Where:

$x(n)$ is the clean signal
$\hat{x}(n)$ is the enhanced signal
N is the frame length (typical choices range between 15ms and 20ms)
M is the number of frames in the signal

This measure can be also extended to the frequency domain (in such case it is usually abbreviated as $fwSNR_{seg}$) The main advantage of using the frequency-based version is the flexibility to place a perceptually motivated frequency spacing as well as to compute different weights for the FFT bins. $fwSNR_{seg}$ formula is shown in (2.4).

$$fwSNR_{seg} = \frac{10}{M} \sum_{m=0}^{M-1} \left(\frac{\sum_{j=1}^{K} W_j log_{10} \left[X^2(j,m) / \left(X(j,m) - \hat{X}(j,m) \right)^2 \right]}{\sum_{j=1}^{K} W_j} \right) \tag{2.4}$$

Where:
W_j is the weight of the j_{th} frequency band
K is the number of bands
M is the total number of frames
$X(j,m)$ is the filter-bank amplitude of the clean signal in the j_{th} frequency band at the m_{th} frame
$\hat{X}(j,m)$ is the filter-bank amplitude of the enhanced signal in the same band and frame

2.6.3 STOI (Short-Time Objective Intelligibility)

STOI is a widely used measure for evaluating speech intelligibility [49]. It is a function of the clean and degraded speech whose output is a scalar value between 0 and 1 that is expected to have a monotonic relation with the average intelligibility (i.e. the percentage of correctly understood words averaged across a group of users). A value close to 1 represents a very intelligible speech and a value close to 0 represents unintelligible speech. Alternatively, these values can be presented as percentages between 0% and 100%. STOI uses a sample rate of 10kHz that is considered to be sufficient to capture the most relevant frequency range related with intelligibility [50]. First, signals are decomposed using a perceptually motivated filterbank. Then, a short time segmentation with 50% overlap, Hanning-windowed frames with 256 samples and zero-padding up to 512 is performed. In order to compensate for global level differences, a normalization and clipping procedure is applied. Finally, a band-wise correlation coefficient is calculated and averaged to return the final score. A diagram of this measure is shown in Figure 5.

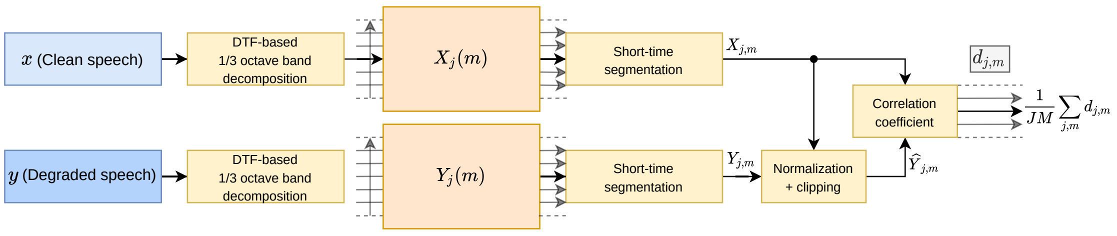

Figure 5: STOI is a function that takes the clean signal x and degraded speech signal y and decomposes them into DFT-based, one-third octave bands $X_j(m)$ and $Y_j(m)$. After a short-time segmentation and normalization plus clipping of the degraded speech, a correlation coefficient $d_{j,m}$ for each band is calculated and averaged.

2.6.4 SI-SDR (Scale-invariant Signal-to-distortion Ratio)

SI-SDR was proposed by Le Roux et al. [51] as a modification of the original signal-to-distortion ratio (SDR) to prevent scaling issues that could artificially increase the obtained SDR value. It is commonly used to evaluate source separation models, and corresponds to a logarithmic comparison between a target signal e_{target} and a residual e_{res} that is calculated as the difference between the target and the estimate $\hat{s}$. The relationship between these three signals can be also summarized as $\hat{s} = e_{target} + e_{res}$. SI-SDR is computed by the formula shown in (2.5).

$$\text{SI-SDR} = 10log_{10}\left(\frac{\|e_{target}\|^2}{\|e_{res}\|^2}\right) \tag{2.5}$$

2.6.5 PESQ (Perceptual Evaluation of Speech Quality)

PESQ [52] was born after a competition held by ITU-T study group 12 [53] to select a new model with good performance across a very wide range of codec and network conditions such as coding distortion, packet loss and background noise. The two algorithms with the highest performance of the competition were combined to generate PESQ. The calculation of this metric involves several stages. First, the reference and degraded signal are aligned to a standard listening level. Using a FFT, they are filtered to model a standard telephone handset and then processed through an auditory transform that involves equalizing for linear filtering and gain variations in the system. After this transformation, two distortion parameters are extracted from what is known as *the disturbance* (the difference between both transformations of the signals). These parameters are finally aggregated in frequency and time, and mapped to predict a mean opinion score (MOS). A high level overview is shown in Figure 6.

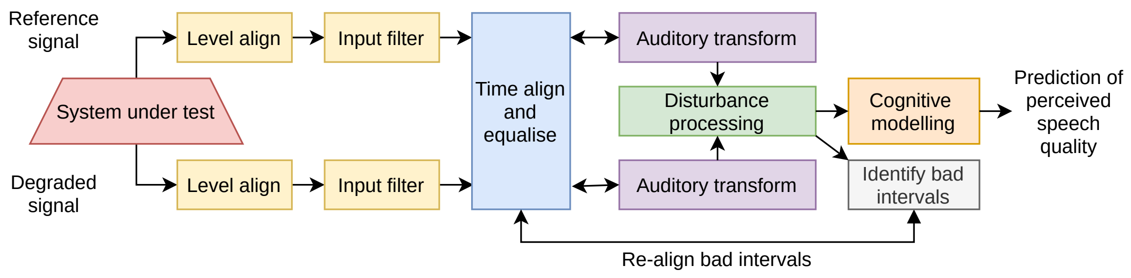

Figure 6: High level overview of Perceptual Evaluation of Speech Quality (PESQ). A reference and degraded signal pair is mapped to a prediction of perceived speech quality.

2.6.6 DNSMOS

DNSMOS [54] is an objective perceptual quality metric designed to evaluate noise suppression algorithms and is based on MOS estimation. It uses a Convolutional Neural Network (CNN) that was trained using ground truth human ratings obtained using the ITU-T P.808 [47], a subjective listening testing framework. At the time of writing this thesis, it is provided as part of the Deep Noise Suppression Challenge as an Azure[1] service for researchers. According to its authors, it has shown around 0.78 and 0.79 correlation with PESQ and POLQA, respectively, when Pearson correlation coefficient is used.

2.6.7 POLQA (Perceptual Objective Listening Quality Analysis)

POLQA [55] is a voice quality testing technology for fixed, mobile and IP based networks. It is a model to predict listening quality as it is perceived in an ITU-T P.800 listening experiment. It uses a psycho-acoustic model to emulate human perception and transform the sound into an internal neural representation. POLQA takes into account the effects of human hearing and the signal is subdivided into short segments, and transformed to the spectral domain. A transformation to the Bark scale is also applied. In the Bark domain, it calculates masking thresholds. The final result of the POLQA pipeline is a mapping into MOS scale. An overview of the algorithm is shown in Figure 7.

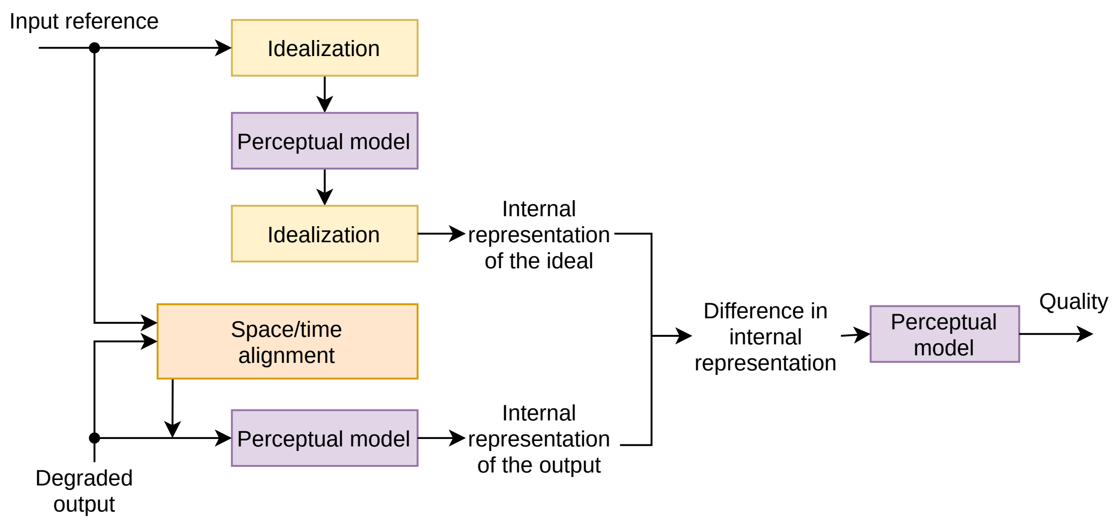

Figure 7: Overview of Perceptual Objective Listening Quality Analysis (POLQA) algorithm.

[1]https://azure.microsoft.com/

2.6.8 ViSQOL (Virtual Speech Quality Objective Listener)

ViSQOL (Virtual Speech Quality Objective Listener) v3 [56] was released as a C++ library used to assess the perceived quality of speech and eliminate the need of requiring a license to use it as in the case of POLQA. ViSQOL was designed with a polynomial mapping of the neurogram similarity index measure (NSIM) of MOS. It uses the reference and degraded signal, performs a global alignment and calculates their gammatone spectrogram. It also features subsequent stages of spectrogram thresholding, voice activity detection (VAD), and patch and subpatch alignment (forms of local alignments). The resulting outcome is a mapping to a MOS score. See Figure 8.

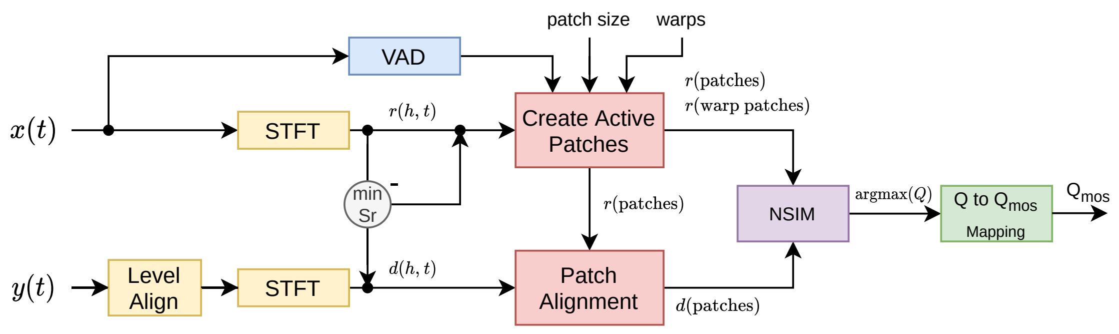

Figure 8: High level overview of Virtual Speech Quality Objective Listener's algorithm (ViSQOL).

2.6.9 WARP-Q

WARP-Q is a full-reference objective speech quality metric that uses dynamic time warping cost for a Mel-frequency cepstral coefficients (MFCC) based representation of the speech signal [57]. It has been designed to be robust to micro-alignment differences present in novel generative speech codecs. These codecs are aimed to compress speech signals at a low bit rate while retaining high speech quality. This procedure could present structural differences between the original and generated signals that may not be perceptually significant to the listener, but cause quality prediction issues when using existing metrics such as POLQA, PESQ or ViSQOL. Regardless of being oriented to speech codecs, WARP-Q is also deemed as a versatile metric for general quality assessment scenarios.

WARP-Q uses voice activity detection (VAD) to discard non-speech segments. Then, a normalized MFFCs representation of the reference and degraded signals are generated using 12 critical bands up to 5kHz, followed by a cepstral mean and variance normalization (CMVN) algorithm. Finally, a subsequence dynamic time warping (SDTW) algorithm is used, and the final score corresponds to the median of align costs between different signal frames. A diagram of the different stages of this metric is presented in Figure 9.

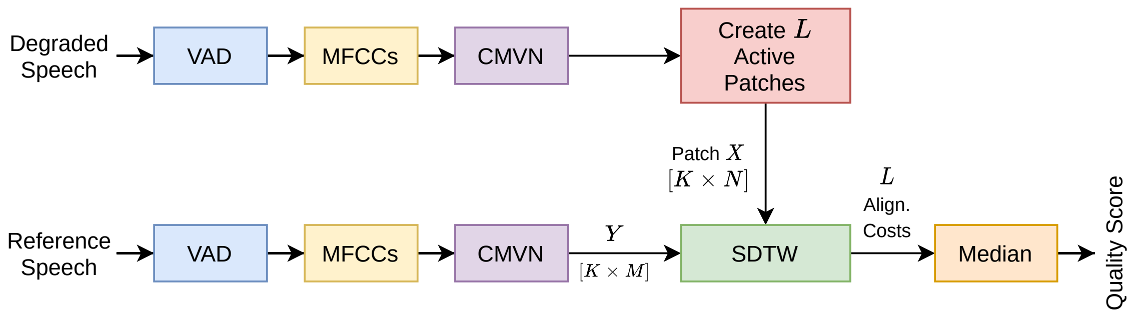

Figure 9: Overview of WARP-Q.

Chapter 3

Methodology

The following section explains the methodology used in the development of this work. It includes the rationale behind the model selection, dataset curation, data augmentation strategies and the evaluation procedures used to characterize each model in three different aspects: Performance, real time efficiency and model complexity.

3.1 Model selection

After each edition of the Deep Noise Suppression Challenge organized by Microsoft, a group of state of the art models for speech enhancement are released with their respective accompanying papers and in some cases, with publicly available code repositories. Additionally, each Deep Noise Suppression Challenge delivers a new pretrained baseline model whose architecture is explained in the challenge's description paper. For the purpose of this work, two models were selected in order to assess their performance as well as to expand upon their baseline architectures. One of the models was the originally provided baseline of the latest challenge (NSNet2). However, during the development of the thesis, the same authors released a model using a different architecture that, according to their latest research, outperformed the previous baseline and therefore, this model (CRUSE) was used in this thesis instead. Since this paper did not include a repository, an open source implementation of this model was developed by the author of this thesis. By choosing a baseline provided by the Deep Noise Suppression Challenge organizers, it would be possible to establish a set of performance capabilities upon which participants are expected to develop and improve their models.

The second selected model (DTLN) was chosen because of the following reasons: first, the authors provided both the research paper and the model's implementation. Secondly, the comprehensive documentation provided all the necessary details to determine that the model was apt to be trained without needing overly expensive hardware. Finally, potential architectural variations of the baseline model could be formulated and the experiments to test these, were able to be carried out along the development of this thesis.

After choosing the baseline models, four additional variants were included, adding up to a total of six trained models. Each variant is described in Chapter 4.

3.2 Datasets

As of the moment of this thesis, the largest publicly available corpus for speech enhancement to the author's knowledge is the DNS-Challenge dataset previously described in Chapter 2. Along with it, a noisy speech synthesizer script is included, making this resource an ideal candidate for the research of this thesis. In order to prevent a biased evaluation of the model on potential training-like data, an additional collection of datasets composed of the WHAMR!, Flickr Audio Caption Corpus and TSP speech dataset is utilized to generate the test set.

The respective authors of the two chosen models used the same training dataset selected for this work. However, the amount of generated data and training strategies are dissimilar between them as well as the reported metrics in each case. In an effort to provide comparable results in terms of performance, real time efficiency and complexity, the same training, validation and test datasets generated in this work have been used across models as well as the set of calculated metrics. Both are described in the following subsections.

3.2.1 Training and validation sets

40 hours of clean and noisy speech pairs (80 hours total) were generated using the data augmentation script provided with the DNS-Challenge dataset. The generation scripts allow the configuration of a target range in terms of signal-to-noise ratio (SNR) and RT60. Additionally, different subsets of the dataset can be included or excluded of the data generation process. For this work, all subsets were included in order to take into account any potential input fed to the network. The SNR was set in the range of -5dB to 15dB, based in the previously used datasets by the original model's authors, as well as the bibliographic support about estimated SNR ranges in scenarios where a speech enhancement algorithm could be potentially used [58]. Regarding RT60, the selected values range from 0.0 to 0.6 seconds, in order to consider situations ranging from having a speaker that is very close to the microphone in an acoustically dry environment to situations where the reverberation is considerably higher, yet possible to encounter in suboptimal situations [59]. The train validation split used in this work is 80% and 20%, respectively.

After all the models and proposed variants were trained, the best 50% performing variants derived from each baseline were retrained with a larger dataset of 500 hours (500 hours of noisy speech and 500 of clean speech) in order to investigate if the models' performance could improve with a larger amount of data. The parameters in terms of SNR and RT60 were preserved as well as the train validation split percentages.

3.2.2 Test sets

Since the DNS-Challenge dataset gathers a collection of subsets derived from several publicly available datasets, additional datasets were included to test the model with unseen data. These are the WHAMR! dataset, the Flickr Audio Caption Corpus and the TSP speech dataset. The content of these datasets is described in Chapter 2. Due to the high volume of data included in the Deep Noise Suppression Challenge dataset, the test set may not have all the same noise classes represented, however, it is considered to be sufficiently diverse for the expected use case of most noise suppression algorithms. Moreover, previous researchers have also used similar approaches during the Deep Noise Suppression Challenge in order to test their models [34], because the DNS-Challenge's repository provides a test set including only noisy speech files to be used with non-intrusive metrics such as DNSMOS. This strategy was adopted by including these additional datasets in order to include intrusive metrics such as PESQ and ViSQOL during model evaluation.

A 4 hours test set (4 hours of clean and 4 hours of noisy speech) was used to test all models, including the retrained models. Three additional datasets of the same amount of time each, were generated maintaining the previously used SNR range, but changing the RT60. The first dataset contains no reverberant speech. The second dataset has a RT60 range from 0.0 to 0.3 seconds, and the last one goes from 0.3 to 0.6 seconds. These additional datasets were exclusively used to further explore the performance of the best performing models under a more segmented range of RT60.

3.3 Evaluation

The evaluation of each model was carried out in three stages: *Performance*, *real time efficiency* and *model complexity*. Each one targets a specific aspect of the model and all of them together are oriented towards providing sufficient information to make a sensible choice for each real time scenario with known specifications in terms of available computational capabilities.

3.3.1 Performance

Six metrics were calculated to evaluate different aspects of the performance of all models in a 4 hours test dataset. Even when this thesis is focused on speech quality, Short-Time Objective Intelligibility (STOI) was utilized to keep track of any potential loss or improvement in terms of speech intelligibility, because this may be a determining factor for some applications such as hearing aids.

Secondly, scale-invariant signal-to-noise ratio (SI-SDR) was used along with WARP-Q and three different metrics that map to a Mean Opinion Score (MOS) by different means described in Chapter 2. These metrics are Perceptual Evaluation of Speech Quality (PESQ), DNSMOS and Virtual Speech Quality Objective Listener (ViSQOL). Even when these metrics can be regarded as highly correlated, preliminary tests showed differences whose magnitude may affect the decision of the most appropriate model for a specific context, therefore in this work, these three metrics are regarded as complementary. More details about the relationship between these can be found in Appendix A.

3.3.2 Real time efficiency

The guidelines of the Deep Noise Suppression Challenge were followed to evaluate whether a certain model is real time capable or not [14]. These can be essentially summarized in two requirements:

1. Given a frame size of time T_F and a hop size T_H used for the model to operate in real time, the sum of $T_F + T_H$ should add to a maximum of 40 milliseconds.
2. The time needed for the algorithm to process a frame of duration T_F should be less than the hop size T_H.

Both requirements have to be met using an Intel Core i5 quad-core processor at 2.4GHz or equivalent. A diagram of the real-time requirements is shown in Figure 10.

In order to calculate how efficient each model is, 1000 performance cycles of the minimum required amount of frames for each model are computed. During each cycle, its duration is measured and stored. Then, the first 10 cycles are dropped as these shown consistently larger inference times compared to the remaining 990. Then, the arithmetic mean of the remaining stored inference times is reported along with its standard deviation. If the model requires more than one frame to perform a forward pass, the total time is divided by the number of required frames. It is

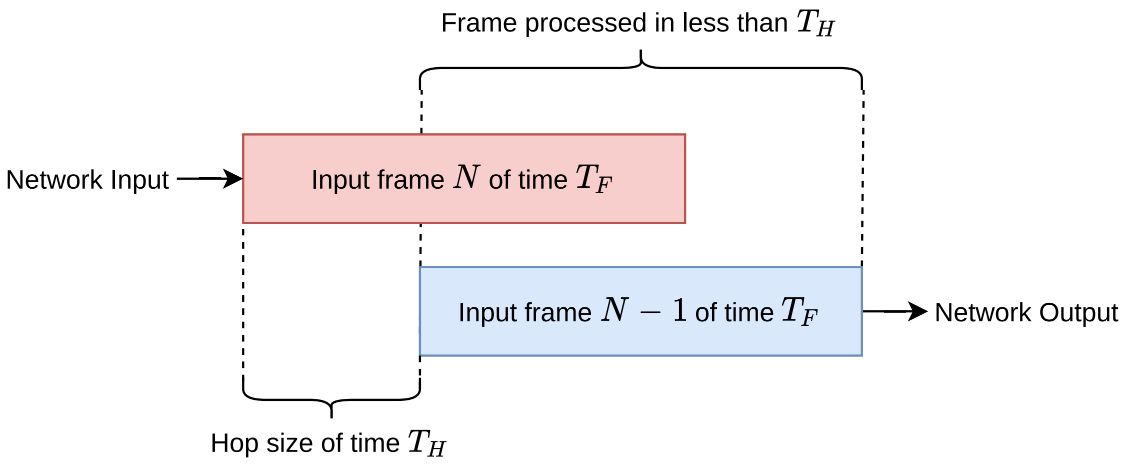
Frame processed in less than T_H
Network Input
Input frame N of time T_F
Input frame $N-1$ of time T_F
Network Output
Hop size of time T_H

Figure 10: Diagram of the factors involved in the real-time requirements guidelines from the Deep Noise Suppression Challenge by Microsoft.

important to mention that this method serves the purpose of being a first referential estimate while choosing the best model for a certain application, rather than an exhaustive way of benchmarking them. Several optimization techniques such as fusing modules, quantizing, pruning the model or even re-implementing it with a different programming language or framework, may impact the accuracy of these results and therefore their appropriateness for a specific scenario.

3.3.3 Model complexity

The model complexity is reported in terms of floating point operations (FLOPs) by using the *ptflops*[1] library. The FLOPs performed by activation functions are not included since the vast majority of the computational burden of each model corresponds to each model's layers. Additionally, some non-linear activation functions could be implemented as approximations whose exact number of FLOPs cannot be easily generalized. FLOPs have been chosen as opposed to multiply-accumulate operations (MACs) since by using these, less assumptions are made about the hardware where a model could be deployed.

[1]`https://github.com/sovrasov/flops-counter.pytorch/`

Chapter 4

Results

This chapter presents the results of the six trained model variants based on the original DTLN and CRUSE architectures. First, each variant is described and then, the results are presented considering three separate aspects (performance, complexity and real time efficiency) as well as the relationships between them that are necessary to consider while choosing a model for a given real-time application.

4.1 Model variants

Six model variants have been proposed. Four of them derive from the DTLN baseline and the remaining two derive from CRUSE.

4.1.1 DTLN variants

The original DTLN along with three variants were investigated. The original separation core structure of the DTLN architecture described in Chapter 2 consist of two stacked LSTM followed by a fully connected layer with a sigmoid activation function. The first variant utilizes stacked GRU instead of LSTM and is henceforth referred to as DTLN GRU. The following two variants are the bidirectional version of the previous ones. DTLN BiLSTM utilizes a bidirectional LSTM stack and DTLN BiGRU utilizes a bidirectional GRU stack. These four variants were selected to explore the tradeoff between performance and complexity brought by the aforementioned changes since all GRU variants contain less learnable parameters compared to its LSTM counterparts. Additionally, bidirectional layers can be used to investigate if this type of information flow through the network may improve its performance. A summary of all DTLN variants is shown in Table 1.

Table 1: DTLN variants.

Model variant	Description
DTLN	The original DTLN model as proposed by [34]
DTLN GRU	Uses a stack of unidirectional GRU layers on each separation core
DTLN BiGRU	Uses a stack of bidirectional GRU layers on each separation core
DTLN BiLSTM	Uses a stack of bidirectional LSTM layers on each separation core

4.1.2 CRUSE variants

The authors of CRUSE originally proposed ten variants [23]. Out of those ten variants, two of them that are suggested to represent an optimal performance given their low complexity compared to the others were selected. The first variant is identified as CRUSEx1GRU and corresponds to a symmetrical convolutional encoder/decoder with an increasing number of channels following the sequence 16, 32, 64, 64 in the encoder and 64, 64, 32, 16 in the decoder. The recurrent layer is comprised of a single GRU. The second variant known as (CRUSEx4GRU) and uses the same channel sequences in both the encoder and the decoder. The only difference between CRUSEx1GRU and CRUSEx4GRU is that the recurrent layer of the latter variant splits the flattened features along the encoder channels into four equal parts, each one feed to one of the four parallel GRUs, allowing a potential parallelization of this implementation in a multi-threaded environment. A summary of these variants in shown in Table 2.

Table 2: CRUSE variants.

Model variant	Description
CRUSEx1GRU	64-64-32-16 encoder, 16-32-64-64 decoder, x1 GRU bottleneck
CRUSEx4GRU	64-64-32-16 encoder, 16-32-64-64 decoder, x4 parallel GRU bottleneck

4.2 Performance

Table 3 shows the results obtained after evaluating each model variant on the test dataset of 4 hours of clean and noisy speech (8 hours total) after being trained on 40 hours of clean and noisy speech pairs (80 hours total). The arithmetic mean and standard deviation of six different metrics described in Chapter 2 are presented.

Table 3: Performance evaluation of different models compared to the metrics obtained for the unprocessed noisy speech after being trained with 40 hours of clean and 40 hours of noisy speech. For each case, the arithmetic mean and standard deviation are reported.

Model	STOI	SI-SDR (dB)	PESQ (MOS)	ViSQOL (MOS)	DNSMOS (MOS)	WARP-Q
Noisy speech	0.84±0.12	8.92±7.39	1.58±0.56	2.57±0.65	2.79±0.33	0.92±0.24
DTLN	0.85±0.11	12.46±5.94	1.91±0.60	2.88±0.67	3.01±0.30	0.90±0.22
DTLN GRU	0.85±0.11	12.38±5.84	1.89±0.60	2.86±0.67	2.98±0.29	0.90±0.22
DTLN BiGRU	**0.87±0.10**	**13.16±5.76**	**2.02±0.64**	**2.99±0.68**	**3.14±0.30**	**0.87±0.21**
DTLN BiLSTM	0.86±0.11	12.92±6.07	1.99±0.64	2.97±0.69	**3.14±0.19**	0.88±0.22
CRUSEx1GRU	0.81±0.12	9.40±4.72	1.75±0.57	2.69±0.61	3.00±0.33	0.95±0.21
CRUSEx4GRU	0.80±0.12	8.31±4.02	1.69±0.49	2.65±0.57	2.98±0.33	0.97±0.20

DTLN BiGRU shows the best overall performance in all six metrics, sharing a similar result with DTLN BiLSTM in the case of DNSMOS. Additionally, all DTLN variants improved the noisy speech intelligibility calculated by STOI, whereas the two CRUSE variants showed a more degraded intelligibility. A similar pattern can be observed in terms of WARP-Q, and SI-SDR in the case of CRUSEx4GRU. All metrics where the mean is lower than the noisy speech are shown in red. Finally, regarding MOS, all models present improvements over the noisy speech.

Considering these results, half of each baseline variants (the best CRUSE variant and best two DTLN variants) in terms of arithmetic mean of the obtained scores were retrained with 500 hours of clean and noisy speech (1000 in total). The results are shown in Table 4.

All metrics show improvements except for DNSMOS in the case of CRUSEx1GRU. Additionally, even when the STOI obtained by CRUSEx1GRU improved, it did not surpass the STOI obtained for the noisy speech. A similar situation is observed for WARP-Q. DTLN BiLSTM presents the best overall results, sharing a similar STOI

Table 4: Performance evaluation of different models compared to the metrics obtained for the unprocessed noisy speech after being trained with 500 hours of clean and 500 hours of noisy speech. For each case, the arithmetic mean and standard deviation are reported.

Model	STOI	SI-SDR (dB)	PESQ (MOS)	ViSQOL (MOS)	DNSMOS (MOS)	WARP-Q
Noisy speech	0.84±0.12	8.92±7.39	1.58±0.56	2.57±0.65	2.79±0.33	0.92±0.24
DTLN BiGRU	**0.88±0.09**	14.06±5.70	2.11±0.62	3.04±0.66	3.17±0.30	0.85±0.20
DTLN BiLSTM	**0.88±0.10**	**14.34±5.90**	**2.23±0.67**	**3.15±0.68**	**3.24±0.31**	**0.83±0.21**
CRUSEx1GRU	0.82±0.11	9.97±4.86	1.80±0.58	2.7±0.61	2.98±0.32	0.94±0.21

with DTLN BiGRU.

4.2.1 Perceptual differences

In addition to the presented numeric results, it is important to highlight that if intermediate predictions are computed (i.e. before the training process is considered complete), perceptual differences suggesting a characteristic pattern for each architecture can be evidenced. These differences are visually supported by Figure 11, where it is possible to observe a remarkable contrast between DTLN and CRUSE variants in the upper half of the spectrum. DTLN variants tend to be less severe in suppressing bins from approximately 4.5kHz upward, as opposed to CRUSE, where some bins are heavily suppressed even if they are present in the target clean speech.

4.3 Model complexity

Table 5 presents the number of trainable parameters and floating point operations (FLOPs) performed by each model along with their training configuration in terms of hop size and window size (equal to the FFT size in all cases) used to train each model. It is important to mention that the STFT is required to train the CRUSE variants, however this is not part of the network architecture and is performed before feeding the data to their input layer. FLOPs are reported using the *ptflops* python library. Activation functions and STFT layers are excluded from this calculation, since this may greatly vary from implementation to implementation and usually do not represent the biggest burden of a given architecture. Subsections 4.3.1 and 4.3.2 present a more detailed analysis of each architecture.

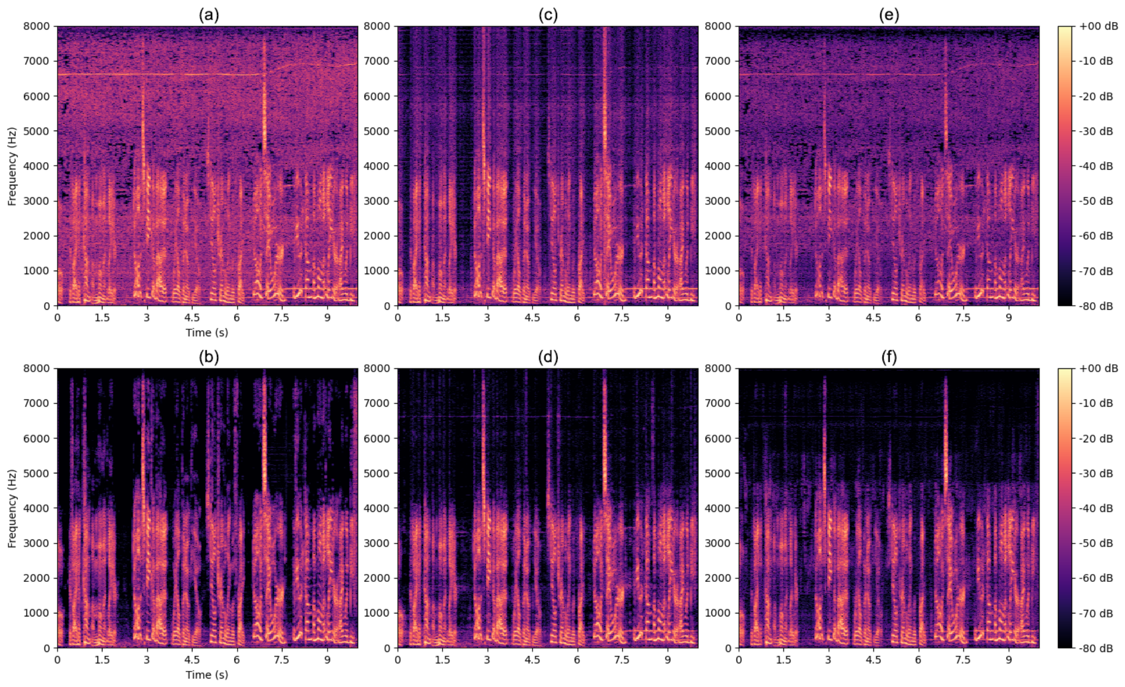

Figure 11: Spectrogram of: (a) Source noisy speech (b) Target clean speech (c) DTLN prediction at an early training epoch (d) DTLN prediction at a nearly complete training epoch. (e) CRUSE prediction at an early training epoch. (f) CRUSE prediction at a nearly complete training epoch.

Table 5: Amount of trainable parameters, STFT configuration and FLOPs performed by each model.

Model	FFT size	Hop size	Trainable parameters	FLOPs
DTLN	512 (32ms)	128 (8ms)	989k	993k
DTLN GRU	512 (32ms)	128 (8ms)	824k	**826k**
DTLN BiGRU	512 (32ms)	128 (8ms)	1.6M	1.5M
DTLN BiLSTM	512 (32ms)	128 (8ms)	1.9M	1.9M
CRUSEx1GRU	320 (20ms)	160 (10ms)	2.1M	6.5M
CRUSEx4GRU	320 (20ms)	160 (10ms)	**518k**	5M

It can be observed that CRUSEx4GRU is the smallest model and CRUSEx1GRU is the biggest one. It is important to consider that CRUSE variants use a smaller FFT size, and therefore the data processed and generated for each forward pass is also a smaller chunk. For instance, if the ratio between the number of trainable parameters and FFT size is calculated, a marginal difference is observed for example between CRUSEx4GRU ($518k/320 = 1618$ trainable parameters per sample) and DTLN GRU ($824k/512 = 1609$ trainable parameters per sample). Furthermore, this ratio is slightly smaller for DTLN GRU. A high correlation between the number of FLOPs and trainable parameters is observed in the DTLN variants. This pattern is not observed in the case of the CRUSE variants.

4.3.1 DTLN variants complexity distribution

Different variants have a distinct number of parameters in each section of its underlying baseline architecture. Figure 12 shows the percentage distribution of trainable parameters in every section of each DTLN variant.

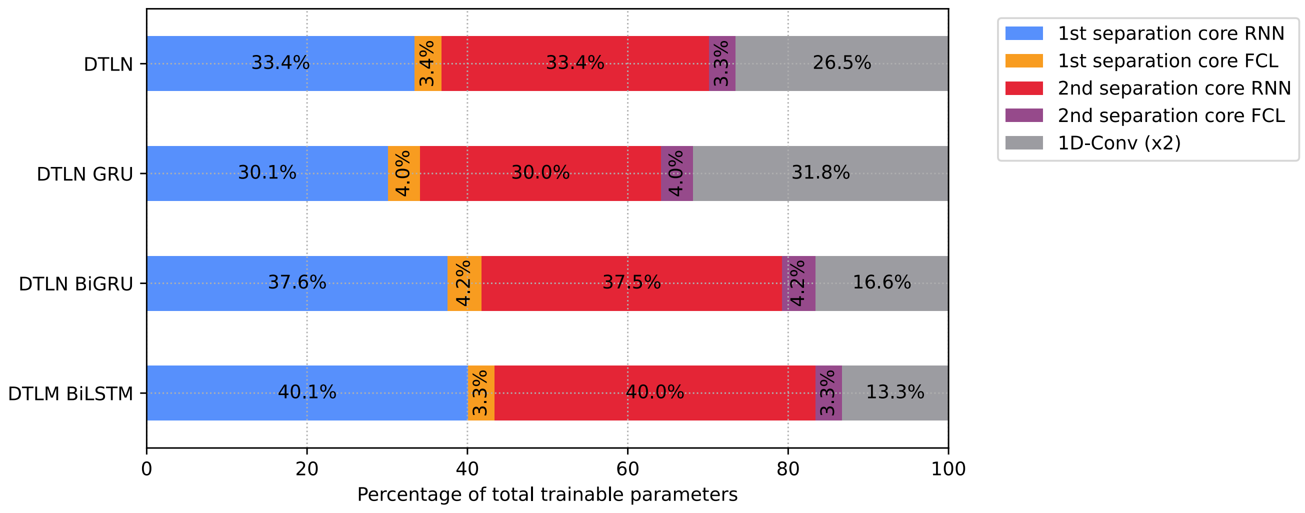

Figure 12: Trainable parameters percentage for each section of each DTLN variant.

In general, the RNN layers of each separation core (i.e. GRU, LSTM, BiGRU or BiLSTM) occupy the largest amount of parameters of each variant, except for DTLN GRU, where the 1D-Conv layers have a slightly higher number of parameters. DTLN BiLSTM is the variant with the largest separation cores.

Figure 13 contains the FLOPs percentage for each section for every DTLN variant. Even when the number of FLOPs is not in all cases the exact same as the number of parameters, these values are highly similar in this architecture. Furthermore, only marginal percentage differences can be evidenced among different sections.

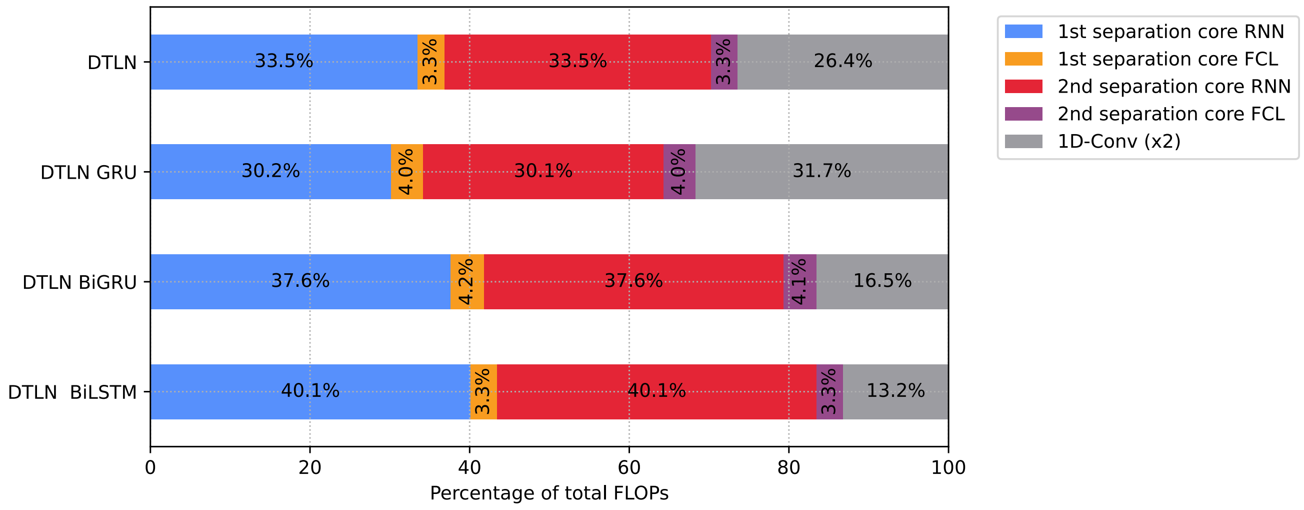

Figure 13: FLOPs percentage performed by each section of every DTLN variant.

4.3.2 CRUSE variants complexity distribution

The distribution of parameters is considerably different in the case of the CRUSE variants. The bottleneck is where most of the parameters of the network are in both cases as it can be seen in Figure 14.

As opposed to DTLN, the distribution of FLOPs shown in Figure 15 is not equally similar with the layers that concentrate the higher amount of parameters. In this case, the decoder would require less storage than the bottleneck, however it performs a significantly larger number of operations, surpassing both the encoder and the bottleneck combined. It is important to notice that the size of the decoder will also depend on the chosen encoder and therefore, choosing the smallest possible encoder would decrease the computational requirements of the decoder too. On the other hand, a smaller bottleneck would imply a smaller memory footprint.

4.3.3 Performance versus complexity

Figure 16 presents the relationship between the complexity of each model and the MOS improvement compared to the noisy speech, assessed by the three utilized metrics that produce a MOS score (i.e. PESQ, DNSMOS and ViSQOL). The x axis

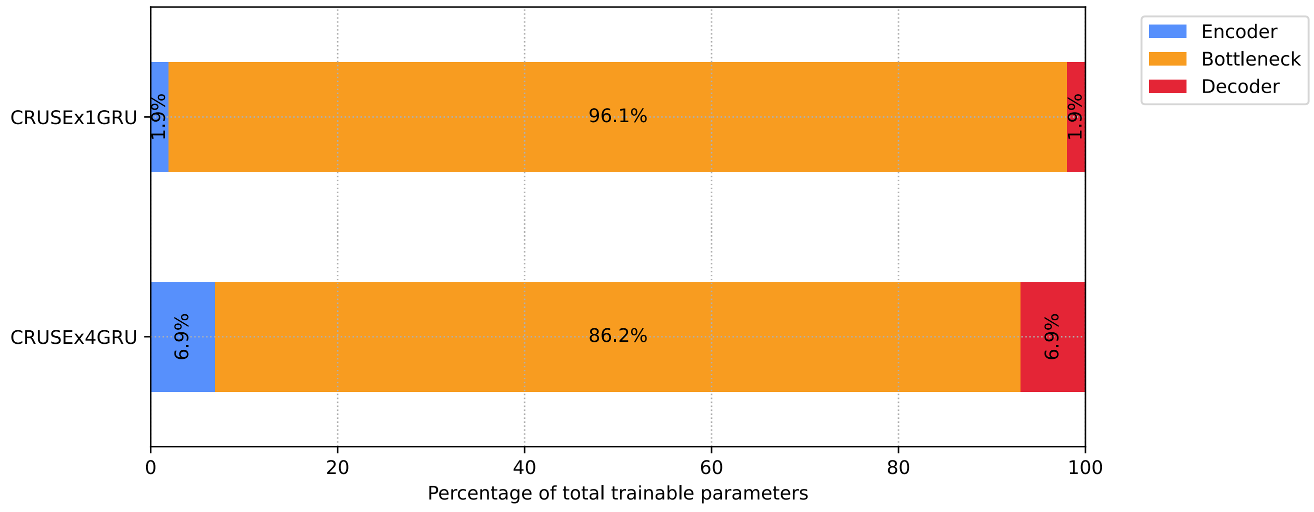

Figure 14: Trainable parameters percentage for each section of each CRUSE variant.

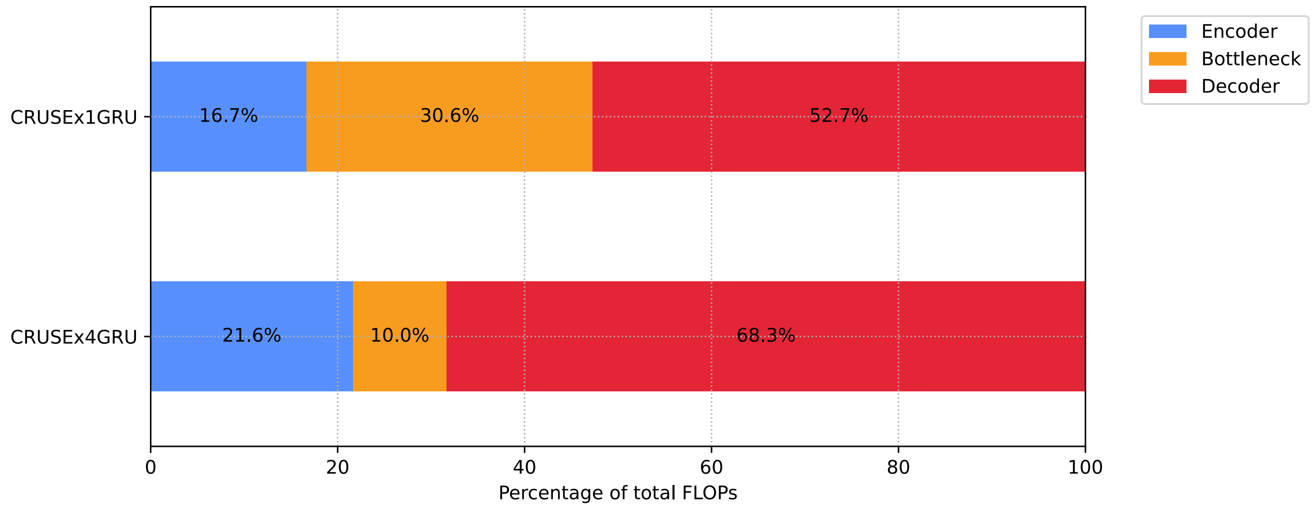

Figure 15: FLOPs percentage for each section of each CRUSE variant.

corresponds to the number of FLOPs required by each model to perform a forward pass. Activation functions and STFT layers are excluded from this calculation. The y axis corresponds to the difference between each MOS value and the MOS of the noisy speech for each corresponding case.

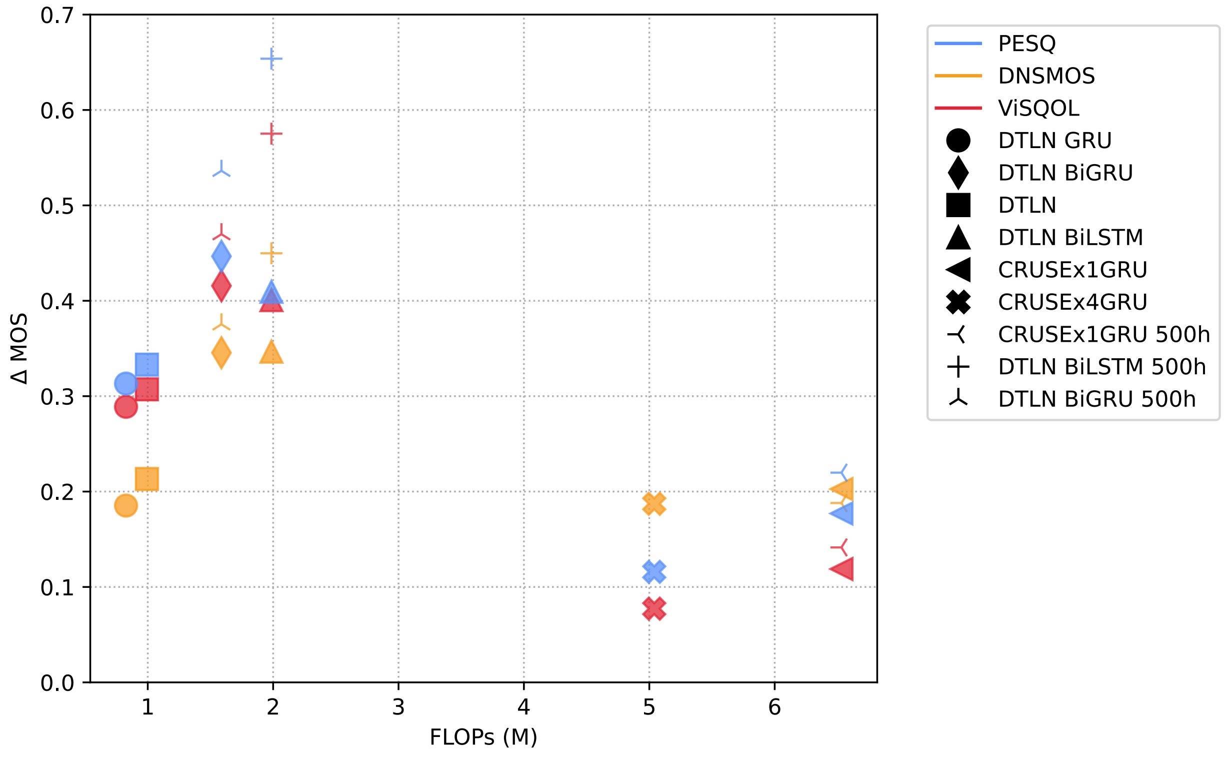

Figure 16: Model complexity in FLOPs versus difference in MOS calculated by PESQ, DNSMOS and ViSQOL compared to the noisy speech (i.e. how much each model improved over the noisy speech).

DTLN variants represent the smaller models that offer the best performance in all metrics. Closely similar results can be observed for CRUSE variants in the case of DNSMOS, however, at a considerably higher number of FLOPs. Within the DTLN variants, DTLN BiLSTM offers the best performance. Furthermore, the performance improved when trained with 500 hours of data. As previously shown in 4.3, the number of trainable parameters is not necessarily correlated with the number of FLOPs as it can be observed for the case of CRUSEx4GRU, that is the model with the least amount of trainable parameters, yet one of the most demanding models in terms of FLOPs. In this case, this difference is mainly caused by the number of operations carried out in the encoder and decoder layers that are not present in the DTLN architecture.

The relationship between the performance of the retrained variants with 500 hours and RT60 was further explored by evaluating them with three additional test datasets of 4 hours each. The first of them has no reverb artificially added, the second one ranges from 0.0 to 0.3 seconds, and the last one from 0.3 to 0.6 seconds. In all cases, the SNR range is the same utilized in all previous evaluations. The results are shown in Figure 17.

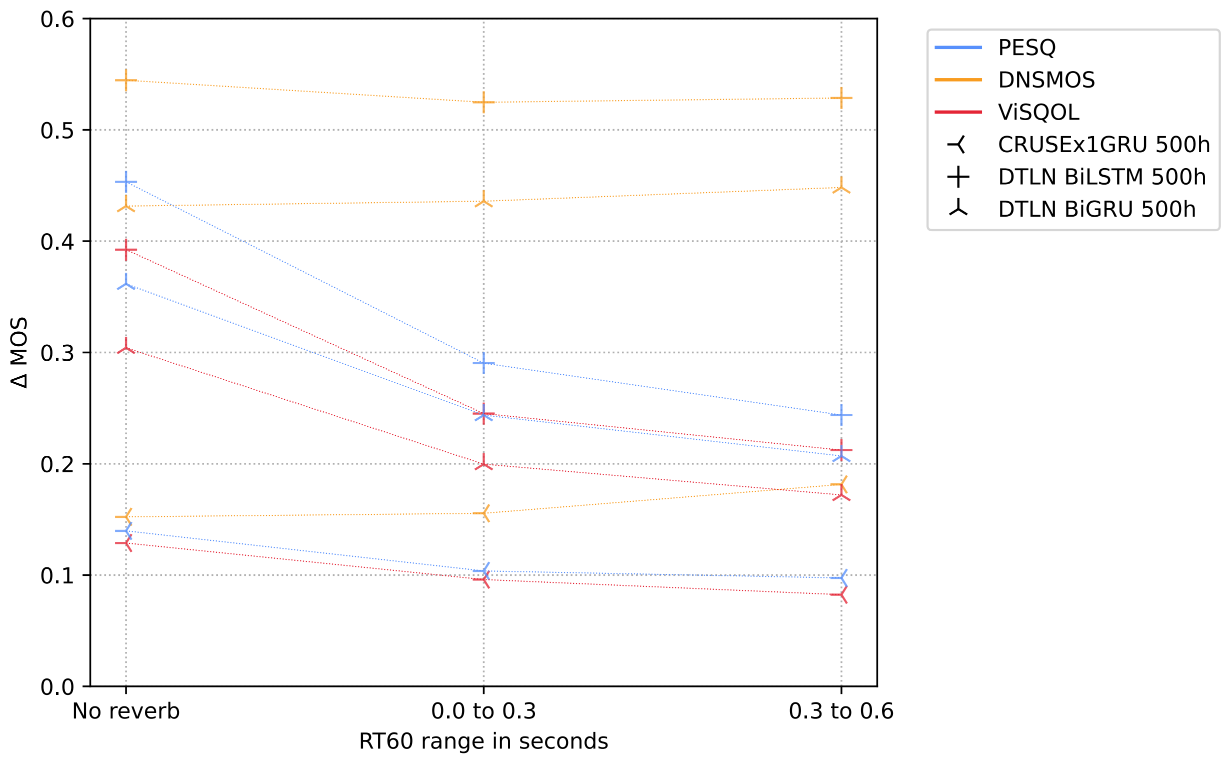

Figure 17: Model complexity in FLOPs vs difference in MOS calculated by PESQ, DNSMOS and ViSQOL compared to the noisy speech with different RT60 ranges.

The results show that the performance in terms of MOS decreases as the RT60 increases for both PESQ and ViSQOL, that also turned out to exhibit the highest correlation among MOS metrics as shown in Appendix A. It is possible to observe the opposite pattern for DNSMOS, however, the increase is negligible and can be regarded as approximately similar instead, suggesting that DNSMOS may remain approximately unchanged upon increasing RT60.

4.4 Real time efficiency

Each variant was evaluated in three different computers below and above the specifications of the Deep Noise Suppression Challenge. One thousand prediction cycles were performed in each case following the procedure explained in 3.3.2. The results are presented in Table 6.

Table 6: Arithmetic mean and standard deviation of the necessary time in milliseconds to complete a forward pass on each model under three different computer specifications.

Model	MacBook Air A1466 i5@1.4GHz	MacBook Pro A1502 i5@2.7GHz	Intel NUC11PAHi7 i7@2.8GHz
DTLN	1.543 ± 0.147 ms	1.345 ± 0.241 ms	0.508 ± 0.011 ms
DTLN GRU	1.455 ± 0.731 ms	1.306 ± 0.250 ms	0.484 ± 0.010 ms
DTLN BiGRU	1.634 ± 0.169 ms	1.615 ± 0.307 ms	0.568 ± 0.009 ms
DTLN BiLSTM	2.459 ± 0.546 ms	2.046 ± 0.318 ms	0.674 ± 0.012 ms
CRUSEx1GRU	0.322 ± 0.045 ms	0.319 ± 0.055 ms	**0.090 ± 0.002 ms**
CRUSEx4GRU	**0.260 ± 0.047 ms**	**0.259 ± 0.061 ms**	0.106 ± 0.004 ms

All models satisfy the real time requirements set by the Deep Noise Suppresion Challenge and have a significant margin that could be used for any potential preprocessing steps or look ahead. If the times in Table 6 are expressed as a percentage of the respective network hop sizes, DTLN variants range from 18.19% (DTLN GRU) to 30.74% (DTLN BiLSTM) for the MacBook Air A1466, thus, leaving a free margin of at least 69.26% of the maximum allowed inference time to be considered real time. In the case of CRUSE, this percentage ranges from 2.60% to 3.22% on the same machine. Overall, CRUSE variants are faster than DTLN variants in all tested machines, however it is important to consider that CRUSE requires the STFT of the input waveform, yet this is not computed inside the network architecture and therefore, this need would increase the required time for a fully functional real time scenario. It is also worth mentioning that the number of parameters are not strictly correlated with the inference time of a given architecture as it can be seen by comparing these results with those of Table 5. As an example, CRUSEx4GRU is the fastest and smaller model, however, CRUSEx1GRU takes less than 0.1 miliseconds more to complete a forward pass even when it is the largest model. Finally, the FFT size used in CRUSE variants is smaller than DTLN variants, hence CRUSE is processing a smaller chunk of audio in each forward pass.

Chapter 5

Discussion and Conclusions

The following section discusses the main results of this research and outlines its main contributions. Finally, potential directions for future work are given along with conclusions summarizing the outcomes of the thesis.

5.1 Real time appropriateness

Several considerations have to be carefully analyzed while choosing the best model for a certain application. There is no single best model or metric that leads to an unequivocal choice. Factors such as memory availability, processing power and application (voice calls, hearing aids, speech recognition, etc) are highly important to take an adequate decision. According to Figure 16, the reader may intuitively think that perhaps DTLN BiLSTM may be the best available model. However, a certain application may have strict memory limitations that may make this model impractical. In such case, a model with a smaller memory footprint such as DTLN or DTLN GRU may be more appropriate. If the goal is to deploy a model in a laptop without significant memory limitations, it is still important to analyze the ecosystem where the model will be run, because in some cases it may not be the only critical process on a given thread. Moreover, in digital audio workstations (DAWs), it is very common to require instancing several copies of a given processor to modify the different signals being recorded in several tracks. Additional considerations and potential optimizations that may impact while choosing a model are discussed in 5.7.

5.2 Weights, biases and FLOPs

It is important to note that the correlation between the size of the parameter tensors and the FLOPs performed by a network is architecture-dependent. This also implies that the complexity of a network is not fully described by only one of these factors. Furthermore and as shown in 4.3.1 and 4.3.2, the weights and biases of some networks such as DTLN, will scale almost identically to their FLOPs, whereas in other architectures such as CRUSE, it does not hold true. As an example, if a new variant of CRUSE has to be created only considering its weights and biases distribution, a researcher could tentatively focus on the bottleneck design since it occupies in all cases more than 85% of the parameters, and overlook the importance of the decoder that performs more than 50% of the FLOPs. Such decision might end up with a model using the optimal size of memory, but with an unexpected suboptimal real time efficiency. Moreover, the interdependence between modules should not be ignored. Regarding the same example, it is possible observe as shown in Figure 15, that the decision about the encoder size has a direct impact on the decoder, and therefore in the number of FLOPs the model has to compute. This is due to the symmetry between encoder and decoder that is present in CRUSE.

5.3 Bidirectional layers

The results obtained in this work support the idea that bidirectional recurrent neural networks can be advantageous within the context of speech enhancement. DTLN BiGRU and DTLN BiLSTM present the best overall metrics as shown in Table 3 an 4. Even when the bidirectional data flow is limited to a single frame for DTLN, overall performance gains are observed compared to their unidirectional variants. A similar approach could be also beneficial for offline applications, where the frame size choice is not constrained by a maximum latency value and could therefore be further increased to investigate if larger sizes maximize performance.

5.4 Data availability

The three selected models to be retrained with 500 hours of speech showed improvements in all metrics with the only exception (and by a marginal difference) being CRUSEx1GRU when evaluated using DNSMOS. In the case of the DTLN variants, the model with the biggest amount of parameters (DTLN BiLSTM in this case) showed also a bigger increase in performance, outperforming DTLN BiGRU which was the variant with the best results when all models were trained with 40 hours. Solely based on this results, it is not possible to support a correlation between the amount of trainable parameters and the improvements that could be achieved by increasing the amount of data. Nevertheless, it is important to emphasize that given a set of trained models, a criteria exclusively based on performance metrics may not be sufficient to separate the best candidate to be retrained when more data is available.

5.5 Impact of perceptual differences

As mentioned in 4.3, some perceptual differences between networks were observed and visual support was found by plotting the spectra of intermediate inference steps of each one of the selected architectures. In the case of CRUSE, such differences would first affect the loss value by creating a bigger difference when compared to the ground truth's, and secondly, they may have direct impact on the resulting quality and intelligibility, specially on the consonant's sounds associated with frequencies above 4kHz in the so-called speech banana [60]. Researchers have also found strong evidence about negative effects of frequencies above 3kHz on patients with noise-induced hearing loss (NIHL), suggesting to raise audiometric thresholds up to around 6kHz instead. These effects are specially impactful on the ability to understand speech when background noise is present [61] and have a direct impact on speech quality as well [62].

5.6 Contributions of this thesis

Three main contributions derived from this work are described in the following subsections.

5.6.1 Model variants comparison

Six variants (including the original baseline architectures) of DTLN and CRUSE are analyzed in terms of performance, complexity and real time efficiency. This evaluation is carried out through a unified procedure where all models are trained and tested on the same respective datasets. This mitigates the frequently encountered issues when different authors train and evaluate their models using rather dissimilar strategies, and therefore making a direct comparison between the obtained metrics insufficient to objectively validate the superiority of a given model. The generated data is also designed in such a way that accommodates the most common encountered scenarios - in terms of SNR and RT60 - where a speech enhancement algorithm could be required, this way, providing a well rounded estimate of a real production setting. Finally, metrics that were not reported by their original authors are also calculated, contributing to extend the analysis of each architecture.

5.6.2 Open source implementations

An open source implementation of CRUSE is released as a product of this research. A DTLN code implementation translation to PyTorch [63] is also included. Since both models (and their respective variants) are available in the same deep learning framework, all possibly different implementation details found across different frameworks are completely eliminated.

5.6.3 Performance improvements

Two out of the six different trained variants (DTLN BiLSTM and DTLN GRU) that are modified versions of their respective original baseline architectures, outperformed them in all the calculated metrics used to evaluate each model's noise reduction performance. Furthermore, an extra performance gain was achieved by increasing the amount of training data.

5.7 Directions of future work

The following section describes a selection of tasks that were not integrally developed throughout this thesis, but that are believed to positively contribute to the goals of this research if investigated further.

5.7.1 Subjective human evaluation

All evaluation metrics used to evaluate the different model variants were automatically computed using different off-the-shelf implementations. As previously explained, three of them map to a MOS score (PESQ, ViSQOL and DNSMOS) and its correlation is further explored in Appendix A. These metrics can be regarded as complementary because choosing a model based only on one of them may lead to different results depending on the metric of choice. Carrying out a subjective human evaluation with trained raters using the ITU-T standard P.808 [64] - similarly to the Deep Noise Suppression Challenge 2021 - is suggested for future work. This process is usually deemed as slower and more expensive than automatic evaluations, however it yields results produced by human listeners. For this reason, this type of evaluation is still considered to be the *gold standard*.

5.7.2 Dereverberation

Even when each variant studied as part of this work was analyzed under several different aspects (performance, model complexity and real time efficiency), there are some facets that still remain unexplored. Particularly, dereverberation is usually treated as a separate task, but could potentially help to enhance speech quality and intelligibility. Preliminary tests showed that the current architectures cannot be directly used for multitasking (for denoising and dereverberation), however, a further analysis of this task is left out of the scope of this thesis, but proposed for future work as a way of improving the current models either by directly modifying their architecture of by combining them with an additional stage while still meeting the real time requirements.

5.7.3 Loss function and performance

There is no current consensus about what loss function is the most suitable in the context of speech enhancement. Previous research has shown evidence that such choice may be dependent on the architecture and that in fact, a loss function that presented the best results in a given scenario, can lead to a poor performance in a different one [65, 66, 67]. Considering this, it is necessary to experiment with different loss functions in order to maximize the model's performance. For instance, experimenting with different loss functions on CRUSE could possibly mitigate the perceptual differences described in 4.3 and discussed in 5.5. Other paradigms such as progressive loss functions could be also explored in future work [68]. Finally, pre-emphasis filtering could be applied to the training data as a way of highlighting the importance of relevant spectral bands for speech quality and intelligibility. This technique has been already successfully used in other audio and deep learning tasks [69].

5.7.4 Model optimizations

CRUSE FFT size

In the case of CRUSE, all variants share the same FFT size used by their original authors (320). While the reason behind this choice is not fully explained in the original paper, such choice may have an impact on the real-time efficiency of any variant of this baseline architecture. More specifically, the STFT can be computed in a more efficient manner if the FFT size is a power 2, delivering a computational complexity of $O(NlogN)$ as opposed to the original $O(N^2)$ complexity of such transform. A detailed description about these algorithms can be found in [70]. These optimizations are also widely used on popular implementations such as $FFTW$ [71].

Other optimizations

Once the most appropriate option has been selected among all trained models, further optimizations could be performed to maximize its efficiency. Some sources of optimization will naturally arise depending on the hardware availability, such as using a GPU to perform parallel processing or single instruction multiple data (SIMD) if a given processor architecture makes it available. Other possible optimizations arise from analyzing the specifics of the architecture of choice. An in depth discussion about this topic is beyond the scope of this thesis, however, some directions for future work are given. First, a network representing an optimal performance could suffer from being over-parametrized (i.e. having more parameters than the minimum required to efficiently perform a given task). In this front, the *lottery ticket hypothesis* can be used to prune a model to identify the most compact representation without harming performance [72]. Secondly, another way of making the model more efficient with a possibly low trade off in performance, is to quantize it. A popular way of achieving this, is by using 8-bit integer numbers (INT8) to represent 32-bit floating point numbers (FP32). This optimization may be specially important for mobile devices or edge devices. A more detailed discussion about this topic is available in [73, 74, 75]. Thirdly, multiple modules can be fused into a single one as explained in [76]. This procedure achieves a more efficient and possibly accurate representation that also saves time spent on memory access. Finally, if the tensors representing the weights and biases of a model are sparse (i.e. having a vast portion of elements equal to zero), it is possible to represent them in a more compact and memory efficient way compared to storing a dense tensor filled mostly with zeros. Sparse coalesced and uncoalesced COO tensors as well as sparse CSR tensors are made available in some popular deep learning frameworks [77, 78]. In some cases, this compact representation is also accompanied by a boost in inference speed [79, 80, 81].

5.8 Conclusions

This work studies the task of speech enhancement on a wide band scenario (monoaural audio sampled at 16kHz). It is mainly focused on improving speech quality by means of deep learning, and oriented toward real-time applications with constrained memory size and processing power. The intersection between speech enhancement and deep learning is currently deemed as a challenging task in its nascent phase. The methodology used in this research attempts to fully characterise the different aspects of a given model involved in determining its appropriateness for a known real-time application. This is achieved by evaluating the model performance, complexity and real-time efficiency and how these aspects are related. It is believed that by proposing a well rounded analysis, researchers, practitioners and enthusiasts can be better informed while having to choose or reject a model for a given real time context.

Two models proposed in the context of the Deep Noise Suppression Challenge by Microsoft were selected as baseline architectures to build upon. A total of six variants of these architectures were trained and studied. Initially, 40 hours of speech were used, and the models were analyzed with six performance metrics (STOI, SI-SDR, PESQ, ViSQOL, DNSMOS and WARP-Q), two different complexity indicators (number of trainable parameters and FLOPs) and their real-time efficiency in terms of inference time under given computational specifications. The best performing models were retrained with 500 hours of speech in order to investigate if their performance could be improved by using a larger amount of data. As a result, all models improved, except for one metric (DNSMOS) that scored lower in one of them. Three additional test sets were used with these models to investigate the relationship between their performance and RT60. The results showed that the performance decreases for higher RT60 ranges in terms of PESQ and ViSQOL. Surprisingly, DNSMOS showed only small variations for different RT60 ranges.

Next, the efficiency versus performance trade off is discussed. Considerations oriented toward choosing the most suitable model for a given application are presented to the reader. Further discussion presents the outcomes of this thesis and its three main contributions: First, all model variants used in this work were analyzed within the proposed unified framework, eliminating any potentially debatable difference that frequently arises when different authors train and test their models on dissimilar data or report different metrics that cannot be easily compared. Secondly, an open source implementation of CRUSE and a PyTorch implementation of DTLN are released. Finally, out of the six analyzed variants, two of them outperformed the previously existing ones in all considered performance metrics. Additionally and as a byproduct of this research, Appendix A contributes to further investigate the correlation between different metrics that map to a MOS.

Future directions that include extending this work with additional tasks that are relevant within the context of speech enhancement (i.e. dereverberation) as well as subsequent procedures that may contribute to pursue a model with maximal performance and minimum computational requirements are outlined.

List of Figures

List of Tables

Bibliography

[1] Loizou, P. *Speech enhancement: Theory and practice*, chap. 1 (CRC Press, 2013).

[2] Coppens-Hofman, M. C., Terband, H., Snik, A. F. M. & Maassen, B. A. M. Speech Characteristics and Intelligibility in Adults with Mild and Moderate Intellectual Disabilities. *Folia Phoniatrica et Logopaedica* **68**, 175–182 (2017).

[3] Weiss, M. R., Aschkenasy, E. & Parsons, T. W. Study and Development of the INTEL Technique for Improving Speech Intelligibility. *Technical Report NSC-FR/4023, Nicolet Scientific Corporation, Northvale, NJ.* (1975).

[4] Boll, S. F. Suppression of Acoustic Noise in Speech Using Spectral Subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **2**, 113–120 (1979).

[5] McAulay, R. J. & Malpass, M. Speech Enhancement Using a Soft-Decision Noise Suppression Filter. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **28**, 137–145 (1980).

[6] Ephraim, Y. & Malah, D. Speech Enhancement Using a Minimum Mean-Square Error Short-Time Spectral Amplitude Estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **32**, 1109–1121 (1984).

[7] Williamson, D. S. & Wang, D. Time-frequency masking in the complex domain for speech dereverberation and denoising. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **25**, 1492–1501 (2017).

[8] Weninger, F. *et al.* Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR. In *12th International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)* (2015).

[9] Xia, Y. *et al.* Weighted Speech Distortion Losses for Neural-Network-Based Real-Time Speech Enhancement. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* **2020-May**, 871–875 (2020).

[10] Wichern, G. & Lukin, A. Low-latency approximation of bidirectional recurrent networks for speech denoising. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 66–70 (2017).

[11] Tan, K. & Wang, D. L. A convolutional recurrent neural network for real-time speech enhancement. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* **2018-September**, 3229–3233 (2018).

[12] Wisdom, S. *et al.* Differentiable consistency constraints for improved deep speech enhancement (2018).

[13] Park, S. R. & Lee, J. A fully convolutional neural network for speech enhancement (2016).

[14] Reddy, C. K. A. R. *et al.* INTERSPEECH 2021 Deep Noise Suppression Challenge (2021).

[15] Zoom Events. URL `https://investors.zoom.us/news-events/events`.

[16] OwlLabs - 10 Key Stats from Owl Labs State of Remote Work 2020. URL `https://resources.owllabs.com/blog/10-key-stats-from-owl-labs-state-of-remote-work-2020`.

[17] Webex blog - 18 video conferencing statistics for 2021. URL `https://blog.webex.com/video-conferencing/18-video-conferencing-statistics-for-2021/`.

[18] Velocity Global - How to Manage a Remote International Workforce.
URL `https://velocityglobal.com/blog/the-complete-guide-how-to successfully-build-and-manage-a-remote-international-workforce/`.

[19] Reddy, C. K. A. *et al.* Interspeech 2021 deep noise suppression challenge. *CoRR* (2021).

[20] Weintraub, M. *A theory and computational model of auditory monaural sound separation*. Ph.D. thesis, Stanford (1985).

[21] Brown, G. J. & Cooke, M. Computational auditory scene analysis. *Computer Speech & Language* **8**, 297–336 (1994).

[22] Xu, Y., Du, J., Dai, L.-R. & Lee, C.-H. An Experimental Study on Speech Enhancement Based on Deep Neural Networks. *IEEE SIGNAL PROCESSING LETTERS* **21** (2014).

[23] Braun, S., Gamper, H., Reddy, C. K. R. & Tashev, I. Towards Efficient Models for Real-Time Deep Noise Suppression (2021).

[24] Feder, M., Oppenheim, A. V. & Weinstein, E. Maximum Likelihood Noise Cancellation Using The Em Algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **37**, 204–216 (1989).

[25] Cooke, M., Green, P., Josifovski, L. & Vizinho, A. Robust automatic speech recognition with missing and unreliable acoustic data. *Speech Communication* **34**, 267–285 (2001).

[26] Kim, G., Lu, Y., Hu, Y. & Loizou, P. C. An algorithm that improves speech intelligibility in noise for normal-hearing listeners. *The Journal of the Acoustical Society of America* **126**, 1486–1494 (2009).

[27] Rickard, S. & Yilmaz, O. On the approximate w-disjoint orthogonality of speech. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 529–532 (2002).

[28] Manilow, E., Seetharman, P. & Salamon, J. *Open Source Tools & Data for Music Source Separation* (2020). URL `https://source-separation.github.io/tutorial`.

[29] van den Oord, A. *et al.* Conditional image generation with PixelCNN decoders. *Advances in Neural Information Processing Systems* 4797–4805 (2016).

[30] van den Oord, A. *et al.* Wavenet: A generative model for raw audio (2016).

[31] Rethage, D., Pons, J. & Serra, X. A wavenet for speech denoising. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing* 5069–5073 (2018).

[32] Lluís, F., Pons, J. & Serra, X. End-to-end music source separation: Is it possible in the waveform domain? *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* 4619–4623 (2019).

[33] Wilson, K. *et al.* Exploring tradeoffs in models for low-latency speech enhancement. *16th International Workshop on Acoustic Signal Enhancement, IWAENC 2018 - Proceedings* 366–370 (2018).

[34] Westhausen, N. L. & Meyer, B. T. Dual-Signal Transformation LSTM Network for Real-Time Noise Suppression (2020).

[35] Luo, Y. & Mesgarani, N. Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation. *IEEE/ACM Transactions on Audio Speech and Language Processing* **27**, 1256–1266 (2019).

[36] Tan, K. & Wang, D. L. Learning complex spectral mapping with gated convolutional recurrent networks for monaural speech enhancement. *IEEE/ACM Transactions on Audio Speech and Language Processing* **28**, 380–390 (2020).

[37] Braun, S. & Tashev, I. Data augmentation and loss normalization for deep noise suppression (2020).

[38] LibriVox | free public domain audiobooks. URL `https://librivox.org/`.

[39] Cao, H. *et al.* CREMA-D: Crowd-sourced Emotional Multimodal Actors Dataset. *IEEE transactions on affective computing* **5**, 377 (2014).

[40] Font, F., Roma, G. & Serra, X. Freesound Technical Demo. In *ACM International Conference on Multimedia (MM'13)*, 411–412. ACM (ACM, Barcelona, Spain, 2013).

[41] Thiemann, J., Ito, N. & Vincent, E. DEMAND: a collection of multi-channel recordings of acoustic noise in diverse environments. *Meetings Acoust.* 1–6 (2013).

[42] Openslr dataset. URL `http://www.openslr.org/resources.php`.

[43] Hershey, J. R., Chen, Z., Le Roux, J. & Watanabe, S. Deep clustering: Discriminative embeddings for segmentation and separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) - Proceedings*, 31–35 (2016).

[44] Scheibler, R., Bezzam, E. & Dokmanic, I. Pyroomacoustics: A python package for audio room simulations and array processing algorithms. *CoRR* (2017).

[45] Harwath, D. F. & Glass, J. R. Deep multimodal semantic embeddings for speech and images. *CoRR* (2015).

[46] Kabal, P. TSP Speech Database URL `http://www.tsp.ece.mcgill.ca/`.

[47] International Telecommunication Union. P.800.1 : Mean opinion score (MOS) terminology. URL `https://www.itu.int/rec/T-REC-P.800.1/en`.

[48] Richards, D. Speech-transmission performance of p.c.m. systems. *Electronics Letters* **1**, 40–41 (1965).

[49] Taal, C. H., Hendriks, R. C., Heusdens, R. & Jensen, J. An algorithm for intelligibility prediction of time-frequency weighted noisy speech. *IEEE Transactions on Audio, Speech and Language Processing* **19**, 2125–2136 (2011).

[50] French, N. R. & Steinberg, J. C. Factors Governing the Intelligibility of Speech Sounds. *Journal of the Acoustical Society of America* **19**, 90–119 (1947).

[51] Le Roux, J., Wisdom, S., Erdogan, H. & Hershey, J. R. SDR-HALF-BAKED OR WELL DONE? URL `http://bass-db.gforge.inria.fr/bss_eval/`. `1811.02508v1`.

[52] Perceptual evaluation of speech quality (PESQ) - A new method for speech quality assessment of telephone networks and codecs. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* **2**, 749–752 (2001).

[53] ITU-T Study Groups (Study Period 2017-2020). URL `https://www.itu.int/en/ITU-T/studygroups/2017-2020/Pages/default.aspx`.

[54] Reddy, C. K. A., Gopal, V. & Cutler, R. DNSMOS: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors. *CoRR* (2020).

[55] Beerends, J. *et al.* Perceptual Objective Listening Quality Assessment (POLQA), The Third Generation ITU-T Standard for End-to-End Speech Quality Measurement Part I-Temporal Alignment. *AES: Journal of the Audio Engineering Society* **61**, 366–384 (2013).

[56] Chinen, M. *et al.* Visqol v3: An open source production ready objective speech and audio metric (2020).

[57] Hines, A., Skoglund, J., Chinen, M. & Jassim, W. Warp-q: Quality prediction for generative neural speech codecs (2021).

[58] K, S., F, W. & M, R. Estimation of signal-to-noise ratios in realistic sound scenarios. *Journal of the American Academy of Audiology* **26**, 183–196 (2015).

[59] NTi Audio. Reverberation Time Measurement URL `https://www.nti-audio.com/en/applications/room-building-acoustics/reverberation-time-measurement`.

[60] Klangpornkun, N., Onsuwan, C., Tantibundhit, C. & Pitathawatchai, P. Predictions from "speech banana" and audiograms: Assessment of hearing deficits in Thai hearing loss patients. *The Journal of the Acoustical Society of America* **134**, 4132–4132 (2013).

[61] Moore, B. C. J. International Journal of Audiology A review of the perceptual effects of hearing loss for frequencies above 3 kHz. *International Journal of Audiology* **55**, 707–714 (2016).

[62] Monson, B. B. *et al.* The perceptual significance of high-frequency energy in the human voice (2014).

[63] Paszke, A. *et al.* PyTorch: An Imperative Style, High-Performance Deep Learning Library (2019).

[64] P.808: Subjective evaluation of speech quality with a crowdsourcing approach. URL `https://www.itu.int/rec/T-REC-P.808-202106-I/en`.

[65] Koyama, Y., Vuong, T., Uhlich, S. & Raj, B. Exploring the best loss function for dnn-based low-latency speech enhancement with temporal convolutional networks (2020).

[66] Kolbaek, M., Tan, Z.-H., Jensen, S. H. & Jensen, J. On loss functions for supervised monaural time-domain speech enhancement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **28**, 825–838 (2020).

[67] Braun, S. & Tashev, I. A consolidated view of loss functions for supervised deep learning-based speech enhancement (2020).

[68] Llombart, J. *et al.* Progressive loss functions for speech enhancement with deep neural networks. *EURASIP Journal on Audio* **2021**, 1 (2021).

[69] Wright, A., Damskägg, E. P., Juvela, L. & Välimäki, V. Real-time guitar amplifier emulation with deep learning. *Applied Sciences (Switzerland)* **10** (2020).

[70] Smith, J. O. Fast Fourier Transforms (FFT). URL `https://ccrma.stanford.edu/~jos/st/Fast_Fourier_Transform_FFT.html`.

[71] FFTW - FFTW Reference. URL `http://www.fftw.org/fftw2_doc/fftw_3.html`.

[72] Frankle, J. & Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *CoRR* (2018).

[73] Wu, H., Judd, P., Zhang, X., Isaev, M. & Micikevicius, P. Integer quantization for deep learning inference: Principles and empirical evaluation. *CoRR* (2020).

[74] Krishnamoorthi, R. Quantizing deep convolutional networks for efficient inference: A whitepaper (2018).

[75] Jain, A., Bhattacharya, S., Masuda, M., Sharma, V. & Wang, Y. Efficient execution of quantized deep learning models: A compiler approach. *CoRR* (2020).

[76] Fuse Modules Recipe — PyTorch Tutorials 1.9.0+cu102 documentation. URL `https://pytorch.org/tutorials/recipes/fuse.html`.

[77] PyTorch 1.9.0 documentation — torch.sparse. URL `https://pytorch.org/docs/stable/sparse.html`.

[78] TensorFlow Core — Working with sparse tensors. URL `https://www.tensorflow.org/guide/sparse_tensor`.

[79] Argueta, A. & Chiang, D. Accelerating sparse matrix operations in neural networks on graphics processing units. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6215–6224 (Association for Computational Linguistics, Florence, Italy, 2019).

[80] Gale, T., Zaharia, M., Young, C. & Elsen, E. Sparse GPU kernels for deep learning. *CoRR* (2020).

[81] Wang, Z. Sparsednn: Fast sparse deep learning inference on cpus. *CoRR* (2021).

Appendix A

MOS estimation metrics

Three different metrics (PESQ, ViSQOL and DNSMOS) that map to a Mean Opinion Score (MOS) were used to evaluate the performance of the trained models. MOS obtained by means of subjective human evaluation are considered to be the *gold standard* to assess speech quality [54]. However, such procedures are deemed as time consuming and expensive, thus, finding a metric that automatically produces a highly correlated score with human evaluation is desirable. Having such metric is a challenging tasks on its own, given the inherent subjectivity found across different groups of subjects. As a consequence, any estimate of this score is affected by this factor too. Figure 18 shows the pairwise Pearson correlation coefficient of the three utilized metrics that map to a MOS. This correlation matrix is the result of 14410 different evaluations. It can be observed that the strongest correlation is between PESQ and ViSQOL, whereas the correlation of DNSMOS with the rest of the metrics is more than 20% lower. As observed in Figure 19, 67.26% of the time DNSMOS delivered the highest MOS for a given predicted speech file. On the other hand, PESQ had a strong tendency (96.78%) to deliver the lowest MOS among the three metrics. These results reaffirm the idea that replicating the *gold standard* is a subjective and difficult task and therefore, complementary evaluations may be necessary while accepting or discarding a model based on a MOS score.

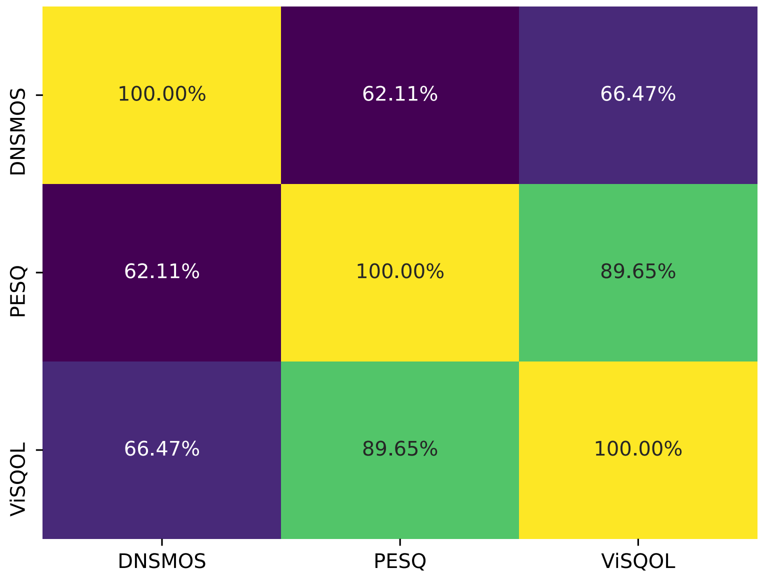

Figure 18: Pearson correlation coefficient between PESQ, ViSQOL and DNSMOS showed as a percentage. The correlation was calculated based on 14410 different evaluations.

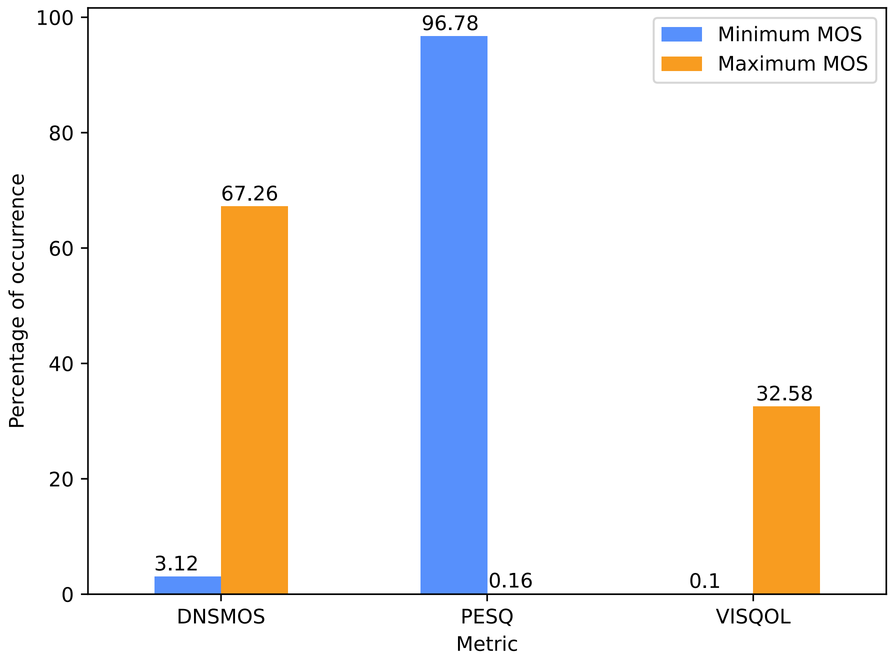

Figure 19: Percentage of occurrence of a given metric delivering the minimum or maximum score among all 14410 performed evaluations.