Exploring Invertible Architecture for Speech Enhancement

M.Tech. Thesis

By
MANSI SINGH

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE
JUNE 2023

Exploring Invertible Architecture for Speech Enhancement

A THESIS

Submitted in partial fulfillment of the
requirements for the award of the degree
of
Master of Technology

by

MANSI SINGH

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE
JUNE 2023

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **Exploring Invertible Architecture For Speech Enhancement** in the partial fulfillment of the requirements for the award of the degree of **MASTER OF TECHNOLOGY** and submitted in the **Department of Electrical Engineering, Indian Institute of Technology Indore**, is an authentic record of my own work carried out during the time period from August 2022 to June 2023 under the supervision of Prof. Vivek Kanhangad, HOD, Department of Electrical Engineering, IIT Indore.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

Mansi Singh
06/06/2023

Signature of the student with date
MANSI SINGH

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

6 June 2023
Signature of the Supervisor of
M.Tech. thesis (with date)
PROF. VIVEK KANHANGAD

Mansi Singh has successfully given his/her M.Tech. Oral Examination held on 15-05-2023.

Signature of Supervisor of M.Tech. thesis
(Prof. Vivek Kanhangad)
Date: 6 June 2023

Convener, DPGC
(Dr. Swaminathan R.)
Date: 08/06/2023

Signature of PSPC Member
(Dr. Sumit Gautam)
Date **06 June '23**

Nagendrakumar
Signature of PSPC Member
(Dr. Nagendra Kumar)
Date: 06 June 2023

Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisor Dr. Vivek Kanhangad for his continuous support and valuable guidance. I would also like to thank Dr. Nitya Tiwari for her help and guidance throughout my work. I have been able to push myself beyond my expectations with their excellent supervision and encouragement.

Besides my advisors, I would like to thank my PSPC members Dr. Sumit Gautam and Dr. Nagendra Kumar for their insightful comments and suggestions towards my research.

I sincerely acknowledge IIT Indore and MHRD for supporting my M.Tech. by providing lab facilities and TA scholarship, respectively.

Last but not least, my work would not have been possible without the encouragement of my parents, friends and fellow lab mates whose tremendous support helped me stay positive and overcome the worst of hurdles.

To them, I shall forever be grateful.

Abstract

Speech enhancement plays a key role in many user-oriented audio applications like telecommunication, assistive hearing, speech recognition, etc. In speech enhancement, the task is to determine the clean speech component from a noisy signal.

The forward process from clean speech to noisy speech is often well-defined, whereas the inverse problem is ambiguous since several parameter sets can map to the same observation i.e., noise can combine with clean speech in different ways to give the same noisy speech signal. In order to address this uncertainty, it is necessary to determine the complete posterior parameter distribution, considering the given measurement. One type of neural network that is particularly suitable for this purpose is known as Invertible Neural Networks (INNs).

INNs focus on learning the forward process, using additional latent output variables to retain crucial information which would have been lost otherwise, while implicitly learning a model of the corresponding inverse process.

Standard ResNet architectures have been made invertible, enabling the same model to be utilized for both generative and discriminative tasks. Invertible ResNets have been shown to perform competitively with state-of-the-art (SOTA) image classifiers and flow based generative models. They also bridge the performance gap between generative and discriminative approaches.

This work explores the possibility of leveraging the i-ResNets for speech enhancement task. This is the first study investigating the applicability of i-ResNets for regression task in general, and speech enhancement in particular, with promising results. The experiments and results on VoiceBank-DEMAND dataset show that the performance is comparable with other related SE models.

Contents

List of Figures

List of Tables

List of Abbreviations

INN	Invertible Neural Network
SE	Speech Enhancement
SOTA	state-of-the-art
STFT	Short Time Fourier Transform
ISTFT	Inverse Short Time Fourier Transform
GAN	Generative Adversarial Network
VAE	Variational Autoencoder
PCA	Principal Component Analysis
ICA	Independent Component Analysis
DNN	Deep Neural Network
SNN	Shallow Neural Network
L-MMSE	Log-Minimum Mean Squared Error
SEGAN	Speech Enhancement Generative Adversarial Network
ANN	Artificial Neural Network
MLP	Multilayer Perceptron
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
ResNet	Residual Network
i-ResNet	Invertible Residual Network
T-F	time-frequency
USL	Unsupervised Loss
ReLU	Rectified Linear Unit
ELU	Exponential Linear Unit
ODE	Ordinary Differential Equation
PESQ	Perceptual Evaluation of Speech Quality
SGD	Stochastic Gradient Descent
MSE	Mean Squared Error

ActNorm	Activation Normalization
BatchNorm	Batch Normalization

Chapter 1

Introduction

1.1 Introduction

The aim of any speech enhancement algorithm is to improve some perceptual aspects of speech, like quality and intelligibility, which have been otherwise degraded by noise. The approach to tackling the overall issue is highly dependent on the specific application, characteristics of the noise source, the nature of the relationship (if any) between the noise and the clean signal, the number of available microphones, etc. Much research has gone in the task of monaural speech enhancement, i.e., speech recorded by a single microphone and it still remains an open problem.

Some of the scenarios in which it is desired to enhance speech include: voice communication over cellular telephone systems which suffers from background noise present in street, car, restaurant, etc. at the transmitter end; persons using hearing aids or cochlear implant devices communicating in noisy conditions; teleconferencing system, present in a noisy environment, broadcasting to other locations etc. Speech enhancement algorithms can be used to pre-process the noisy signal prior to amplification or transmission in such cases.

One of the major issues in speech enhancement task is that noise can be non-stationary, such as the restaurant noise. The spectral and temporal characteristics of non-stationary noise are constantly changing. Also, it need not be additive but can combine with the clean speech signal via convolution or any other non-linear function.

Different types of signal processing based speech enhancement techniques have been developed in the past years such as spectral subtraction, Wiener filtering, minimum mean square error, etc [11]. Many of these approaches to enhance single-channel speech signals are based on the short-time Fourier transform (STFT).

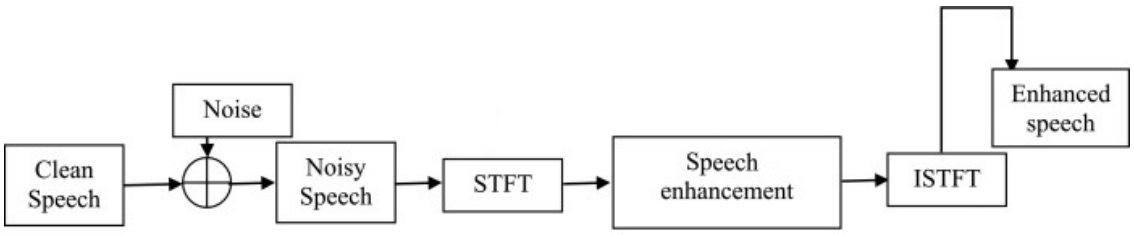

Figure 1.1: Basic speech enhancement framework in the Fourier domain

A general STFT-based speech enhancement procedure is presented in Figure 1.1. The classical signal processing methods make assumptions on the nature of noise such as noise being additive and correlated with the clean speech signal which is not the case in complex real-world scenarios. They also tend to introduce additional noisy artefacts which is a major drawback of these traditional techniques.

Neural network based methods, on the other hand, are purely data driven and do not make any assumptions on the nature and characteristics of the noise or speech signal. Deep learning is a part of machine learning that extensively leverages insights from the human brain, statistics, and applied mathematics. In recent years, it has witnessed tremendous growth in its popularity and usefulness, which can be attributed to advancements such as computers with more robust computing capabilities, larger datasets, and techniques to effectively train deeper neural networks.

Deep learning methods have been shown to outperform the conventional speech enhancement techniques. It is expected to lead to further improvements in terms of accuracy and intelligibility in the future.

1.2 Motivation

Recently, deep neural network models have been shown to yield promising results on speech enhancement task. Several audio generation models that aim to estimate the distribution of the clean speech signal have also been employed directly or with slight adjustments to perform speech enhancement, such as generative adversarial networks (GAN), autoregressive models, variational autoencoders (VAE), and flow-based models.

Traditional approaches typically involve discriminative methods in either the time-frequency (T-F) domain or time domain to learn the mapping from noisy to clean signals. To solve the ill-posed inverse problem of speech enhancement, it is proposed to use Invertible Neural Networks as a tool and employ the most successful discriminative architecture ResNets as the backbone.

Invertible ResNets have demonstrated competitive performance compared to state-of-the-art image classifiers and flow-based generative models, with a single architecture [2]. Consequently, this leads to the assumption that the backbone architecture can be modified for regression tasks and enhancement of speech samples can also be performed.

1.3 Organization of the Thesis

The rest of this thesis is organized as follows: In Chapter 2, the literature survey consisting of different state of the art techniques for speech enhancement is presented along with the problem formulation. Further, in chapter 3, Invertible neural networks and the backbone architecture of ResNet and i-ResNet is discussed. Chapter 4 presents the proposed model, the methodology and the evaluation metric used. In Chapter 5, the experiments and results are discussed in depth. Finally, Chapter 6 includes the conclusion and scope of the future work of this thesis.

Chapter 2

Literature Review and Problem Formulation

2.1 Literature Review

Different types of signal processing based speech enhancement algorithms have been developed in the past years. These algorithms can be divided into the following classes: spectral subtractive algorithms [3], statistical-model based algorithms, like Wiener filtering [10] and minimum mean square error [4], subspace algorithms [5], and binary mask algorithms.

Spectral subtraction technique is the simplest to implement. It estimates the power spectrum of the noise and subtracts it from the observed noisy speech spectrum to enhance the speech signal. The algorithm proposed by Boll [3] in the Fourier transform domain removes the spectral noise bias calculated during non-speech activity to reduce stationary noise in speech.

In statistical-model based algorithms the objective is to determine a linear or nonlinear estimator that can accurately estimate the parameter of interest, e.g., the transform coefficients of the clean signal, given a set of measurements, say the Fourier transform coefficients of the noisy signal. The Wiener filter is a popular method that estimates the clean speech signal by minimizing the mean-square error between the noisy speech and the estimated speech.

Sub-space based methods exploit the fact that the clean speech and noise occupy different subspaces. By estimating the signal and noise subspaces, they can separate and enhance the speech components. Examples include principal component analysis (PCA), and independent component analysis (ICA).

The binary mask algorithms counter the issue of improving speech intelligibility by applying binary gain functions or channel selection where the channels (bands/frequency bins) that satisfy a specific criteria are retained, while channels that fail to meet the criteria are discarded. The first instance of how binary mask algorithms could enhance speech comprehension under monaural conditions was

described in [9].

Due to assumptions on the nature of the underlying noise, such as stationary noise, additive noise, uncorrelated noise, most of these approaches often fail in the complex real-world scenarios. A major drawback of these traditional techniques is the introduction of noisy artefacts in the enhanced signals due to inaccurate spectrum estimation.

Increase in computing capabilities of the computer systems and the availability of large amount of data prompted researchers to look into machine learning for solving the speech enhancement problem. In [17] and [16], connectionist models and analysis of the neural network for noise reduction have been described by Shin'ichi Tamura and Dr. Alex Waibel. One of the earliest application of machine learning for speech enhancement, it is a four-layered feedforward neural network shown to learn the mapping between the noisy and clean signal pairs correctly.

The authors in [19] proposed a regression-based speech enhancement framework using deep neural networks (DNNs) with a multiple-layer architecture, it was an improvement over existing SNN and L-MMSE methods. The DNN based approach offers a strong modelling capability for estimating the complicated nonlinear mapping from observed noisy speech to desired clean speech signals and good generalization on account of a large training set, which might be further enhanced by utilising more noise types in training.

Generative Adversarial Networks operating in time-domain have been used for speech enhancement in [13]. Contrary to earlier works which used spectral domain and/or exploited some higher-level feature, SEGAN processed raw speech and performed better than classic SE methods like Wiener filtering.

EHNET has been proposed by Han Zhao et al in [20]. The model employed both convolutional and recurrent neural network architectures to exploit local structures in both the spatial and temporal domains. This combination resulted in a good trade-off between removing background noise and preserving the real speech signals.

Based on the outstanding modeling capability of diffusion models on natural images and raw audio waveforms, Yen-Ju Lu et al. proposed a diffusion probabilistic model-based speech enhancement (DiffuSE) framework in order to recover clean speech from noisy data [12].

2.2 Problem Formulation

A particular class of neural networks called Invertible Neural Networks (INNs) is well suited for solving the ambiguous inverse problems. Deep residual networks are one of the most successful feed-forward architectures for discriminative learning but they are very different from their generative counterparts. This divide makes it hard for discriminative tasks to benefit from unsupervised learning. This gap can be bridged with a new class of architectures that perform well in both domains. For instance, reversible networks which operate within the same model paradigm and have produced competitive performance on discriminative and generative tasks independently [2]. Noisy to clean speech signal mapping is inherently an inverse problem. So, speech enhancement has been formulated as an inverse, regression based problem and i-ResNet has been utilised for its solution.

Chapter 3

Invertible Residual Networks

3.1 Introduction

Deep learning techniques are based on Neural Networks or Artificial Neural Networks (ANNs), which are a subset of machine learning. ANNs are computational models inspired by the structure and function of neurons found in the human brain. They consist of interconnected nodes, known as artificial neurons or units, organized into layers. These layers typically include an input layer, one or more hidden layers, and an output layer. The connections between neurons, called weights, determine the strength of information flow within the network. Neural networks learn from input data by adjusting these weights during the training process using backpropagation algorithms.

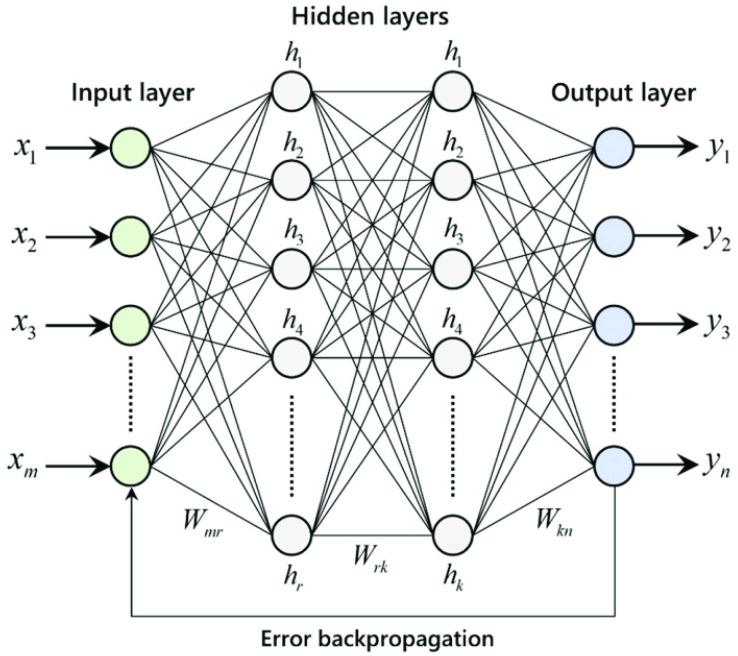

Figure 3.1: Architecture of multilayer ANN with error backpropagation [6].

A neural network with three or more layers qualifies as a deep neural network. Modern deep learning models use tens or hundreds of successive layers to learn hierarchical representations of data. Each layer in the network progressively extracts higher-level features from the input data, allowing for more complex and abstract representations to be learned.

The power of neural networks lies in their ability to learn and generalize from large amounts of data to learn and improve their accuracy over time.

Neural networks can be classified into different types, the most common types include: the perceptron, feedforward neural networks, or multi-layer perceptrons (MLPs), convolutional neural networks (CNNs), recurrent neural networks (RNNs), generative adversarial networks (GANs) etc.

Convolutional Neural Networks: CNNs, or Convolutional Neural Networks, are a specialized type of neural network architecture designed for processing and analyzing structured grid-like data, such as images or time series data. The key feature of CNNs is their ability to automatically learn hierarchical representations of input data through the use of convolutional layers. These layers consist of multiple learnable filters or kernels that slide or convolve across the input data, extracting local features or patterns. This allows CNNs to capture spatial and temporal relationships present in the data.

CNNs typically comprised of multiple layers, including convolutional layers, pooling layers, and fully connected layers. Convolutional layers perform the convolution operation to extract local features, pooling layers downsample the feature maps to reduce computational complexity, and fully connected layers connect the extracted features to the final output layer for classification or regression.

A number of variant CNN architectures have emerged, some of these include: AlexNet, VGGNet, GoogLeNet, ResNet, etc.

3.2 Invertible Neural Networks [1]

The term "Invertible Neural Networks" (INNs) refers to neural network architectures that are invertible by design. They often have tractable algorithms to compute the inverse map and the Jacobian determinant. INNs are bijective function approximators which have a forward mapping

$$F_\theta : \mathbb{R}^d \to \mathbb{R}^d \tag{3.1}$$

$$x \rightarrowtail y \tag{3.2}$$

and an inverse mapping

$$F_\theta^{-1} : \mathbb{R}^d \to \mathbb{R}^d \tag{3.3}$$

$$y \rightarrowtail x \tag{3.4}$$

INN can map both input data samples to targets and also recover the original input data samples from the predictions, thus allowing for bijective mappings. An INN is expected to have the following three properties:

1. Bijective mapping from inputs to outputs such that its inverse exists,
2. Efficiently computable forward and inverse mapping, and
3. Tractable Jacobian for both mappings to allow explicit computation of posterior probabilities.

INNs have many machine learning applications such as probabilistic modeling, generative modeling, representation learning, feature extraction, and solving inverse problems.

The noise interacts with clean signal in different ways to generate noisy signal. Often, the forward process is well-defined, whereas the inverse problem is ambiguous; parameter sets (clean speech, noise) can interact additively, via convolutions or any other non linear function to result in noisy measurement.

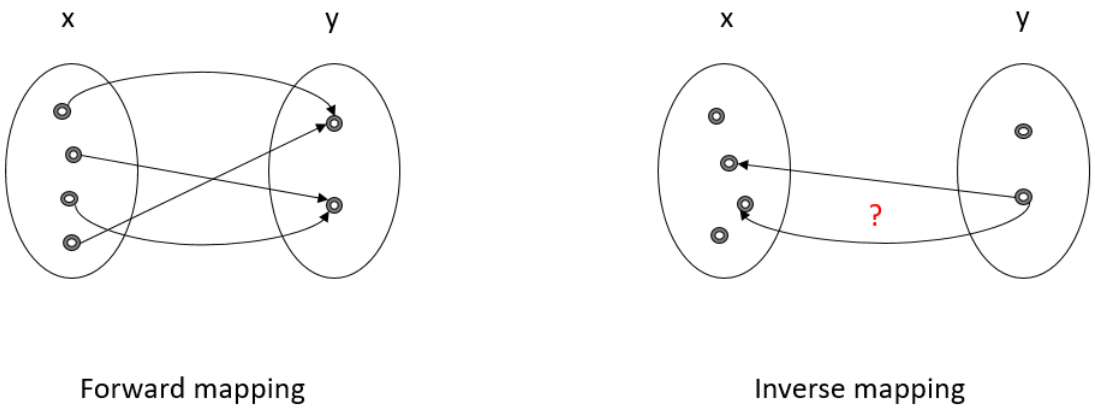

Figure 3.2: Ambiguity of Inverse problem

DNNs try to learn the mapping from noisy to clean which is the inverse problem. Their performance is limited in real life noisy situations because real life noise need not be additive. To overcome this limitation, we explore invertible neural networks which are well suited for solving inverse problems. Networks that are invertible by construction have the advantage that they can be trained on the well-understood forward process x $\rightarrow$ y and the inverse process y $\rightarrow$ x can be implicitly obtained by a backwards pass at the prediction time.

To compensate for the intrinsic information loss during the forward process, INN introduces additional latent output variables z to retain the information about x that is not contained in y. Thus, hidden parameter values x are associated with unique pairs [y, z] of observed measurements and latent variables. During forward training, the mapping $[y,z] = f(x)$ is optimised and its inverse $x = f^{-1}(y,z) = g(y,z)$ is implicitly determined. The density p(z) of the latent variables is modeled as a Gaussian distribution. Thus, the INN provides the desired posterior $p(x|y)$ by a deterministic function $x = g(y,z)$ that transforms the known distribution p(z) to x-space, conditioned on y.

INN differ from the standard approach in using a supervised loss exclusively for the well-defined forward process x $\rightarrow$ y. On the other hand, an unsupervised loss (USL) is utilized to ensure that the generated x adheres to the prior distribution p(x), and the latent variables z are made to follow a Gaussian distribution, also by an unsupervised loss.

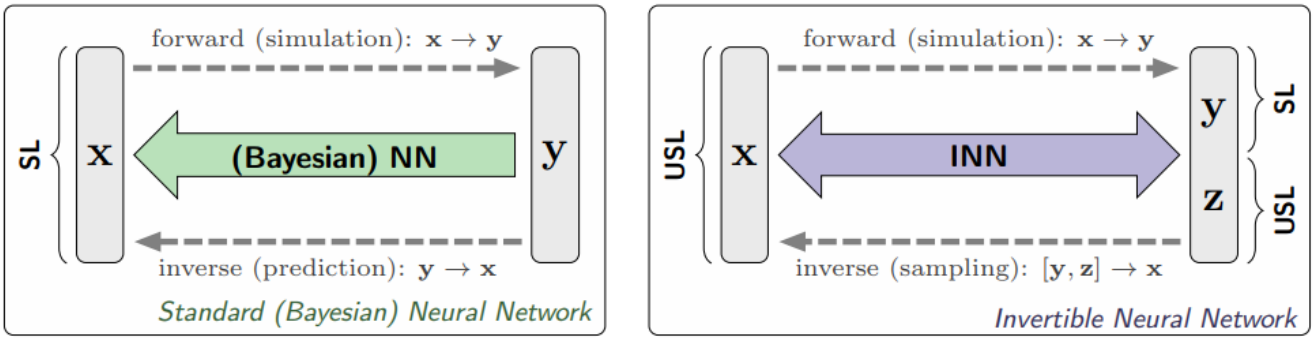

Figure 3.3: Abstract comparison of standard approach (left) and INN approach (right) [1]

By using invertible operations, INN allows the propagation of gradients through both the forward and inverse path during training. This property is useful in tasks such as generative modeling and data synthesis, where the ability to generate samples from learned distributions is required.

3.3 Residual Neural Network [8]

ResNet is a deep learning model in which the weight layers are designed to learn residual functions with reference to the layer inputs. It has a convolutional neural network (CNN) architecture designed to support hundreds or thousands of convolutional layers. It is realized by feed-forward neural networks with "shortcut connections" to address the problem of vanishing gradient as the network gets deeper. Shortcut connections in ResNets bypass one or more layers by performing identity mapping. The outputs from these connections are then added to the outputs of the stacked layers.

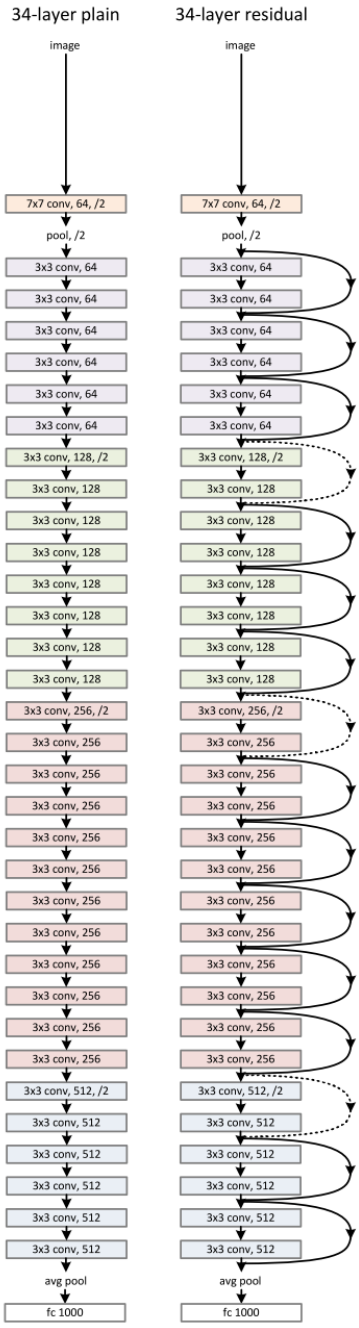

Figure 3.4: A plain network (left) and a residual network (right) with 34 parameter layers [8]

The main building block of ResNet is the residual block. A residual block consists of multiple convolutional layers, followed by batch normalization and ReLU activation functions. There could be two types of residual blocks: the identity block and the projection block.

- **Identity block:** It has the same input and output dimensions, and consists of two or three convolutional layers, with batch normalization and ReLU activation applied after each convolution.
- **Projection block:** The input and output of the block have different dimensions. It includes a convolutional layer with stride 2 to downsample the spatial dimensions of the input, followed by batch normalization and ReLU activation. Downsampling helps in matching the dimensions of the input and output for the skip connection.

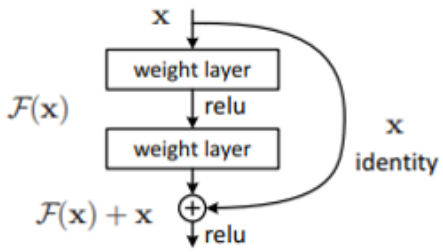

Figure 3.5: A Residual Block [8]

Let $F(x)$ be the underlying mapping to be fit by a number of stacked layers, with x being the input to the first layer. We assume that if multiple nonlinear layers can asymptotically approximate complicated functions, then it follows that they can gradually approximate the residual function, i.e., $F(x) - x$ (assuming that the input and output have the same dimensions). So instead of approximating $F(x)$, we explicitly let these layers approximate a residual function $G(x) = F(x) - x$. The original function then becomes $F(x) = G(x) + x$.

The degradation problem suggests that approximating identity mappings using multiple nonlinear layers can be challenging. This reformulation affects the ease of learning as optimizing the residual mapping is easier than optimizing the original, unreferenced mapping. With the residual learning reformulation, if identity mappings are optimal, simply reducing the weights of the multiple nonlinear layers to zero will result in achieving identity mappings.

3.4 Invertible Residual Network [2]

Simple modifications in the standard ResNet architectures make them invertible. Invertible ResNets (i-ResNets) can be constructed by simply changing the normalization scheme of standard ResNets by adding a simple normalization step during training, already available in standard frameworks. This enables unconstrained architectures for each residual block, with the condition that the Lipschitz constant is smaller than one for each block. Using the remarkable similarity between ResNet architectures and Euler's method for ODE initial value problem

$$x_{t+1} \leftarrow x_t + g_{\theta_t}(x_t) \tag{3.5}$$

$$x_{t+1} \leftarrow x_t + h f_{\theta_t}(x_t) \tag{3.6}$$

where $x_t \in \mathbb{R}^d$ represent states, t represents layer indices or time, h > 0 is a step size, and g_{θ_t} is a residual block. By Picard's Existence and Uniqueness Theorem for first order differential equations, if the function f_{θ_t} is Lipschitz continuous then the solution to ODE exists and is unique [7].

Layers of i-ResNet can be interpreted as a discretization of ordinary differential equation

$$x_{t+1} - x_t = g_{\theta_t}(x_t) \tag{3.7}$$

$$\frac{dx_t}{dt} = \lim_{\delta_t \to 0} \frac{x_{t+\delta_t} - x_t}{\delta_t} = g_{\theta_t}(x_t) \tag{3.8}$$

Solving the dynamics in (3.5), (3.6) backwards in time gives the implicit backward Euler discretization and would result in an implementation of an inverse for the corresponding ResNet.

$$x_t \leftarrow x_{t+1} - g_{\theta_t}(x_t) \tag{3.9}$$

$$x_t \leftarrow x_{t+1} - h f_{\theta_t}(x_t) \tag{3.10}$$

Sufficient condition: Let $F_\theta : \mathbb{R}^d \to \mathbb{R}^d$ with $F_\theta = (F_\theta^1 \circ ... \circ F_\theta^T)$ denote a ResNet with blocks $F_\theta^t = I + G_{\theta_t}$. Then, the ResNet F_θ is invertible if

$$Lip(G_{\theta_t}) < 1, \tag{3.11}$$

for all t = 1, . . . , T, where $Lip(G_{\theta_t})$ is the Lipschitz-constant of G_{θ_t}

This condition is sufficient but not necessary for invertibility.

Let $F(x) = G(x) + x$ be the output from a residual layer and initialise y:=F(x), then inverse of i-ResNet can be obtained via fixed-point iteration:

Algorithm 1. Inverse of i-ResNet layer via fixed-point iteration.

Input: output from residual layer y, contractive residual block g, number of fixed-point iterations n
Init: $x^0 := y$
for $i = 0, \ldots, n$ **do**
  $x^{i+1} := y - g(x^i)$
end for

Figure 3.6: Algorithm for Inverse of i-ResNet layer [2]

Lipschitz constants: Let $F(x) = G(x) + x$ with $Lip(G) = L < 1$ denote the residual layer. Then, it holds

$$Lip(F) \leq 1 + L \tag{3.12}$$

and

$$Lip(F^{-1}) \leq \frac{1}{1-L} \tag{3.13}$$

Hence, stability is guaranteed by design in invertible ResNets for both the forward and inverse mapping.

Residual blocks are implemented as a composition of contractive nonlinearities ϕ (e.g. ReLU, ELU, tanh) and linear mappings. They show excellent performance in both discriminative and generative modeling, with one unified architecture. Using i-ResNets for tasks which require invertibility or semi-supervised learning is a promising approach.

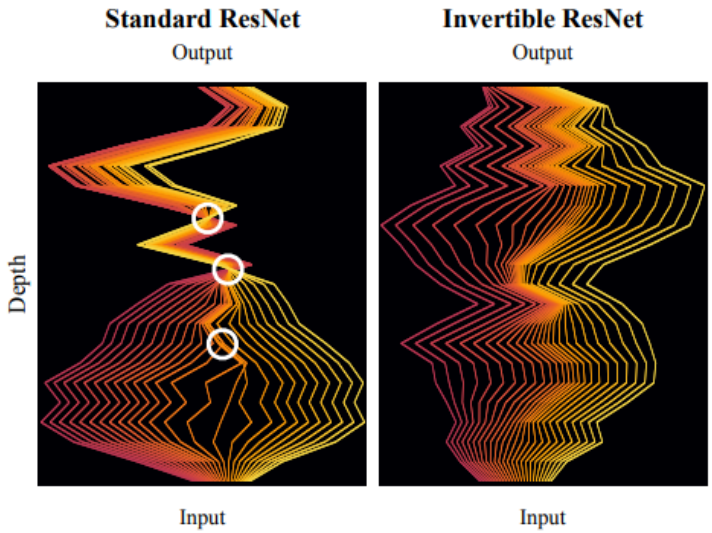

Figure 3.7: Standard residual network (left) and invertible residual network (right) [2]

3.5 Generative Modeling with i-ResNets [2]

Generative models differ from discriminative models in that instead of learning direct mappings from noisy to corresponding clean speech target data, they aim at learning the inherent properties of speech, such as its spectral and temporal structure, by learning a prior distribution over clean speech samples.

This knowledge about the prior is used to make inferences about clean samples given noisy signals that may even lie outside the learned distribution.

The i-ResNet offers a key advantage in designing models that can learn the underlying distribution of a given dataset and generate new samples from that learned distribution because of its ability to perform exact likelihood estimation.

To define a simple generative model for data $x \in \mathbb{R}^d$, first $z \sim p_z(z)$ is sampled where $z \in \mathbb{R}^d$ then x is defined as $x = \phi(z)$ for some function $\phi : \mathbb{R}^d \to \mathbb{R}^d$. If ϕ is invertible and $F = \phi^{-1}$, then the likelihood of any x under this model can be computed using the change of variables formula

$$p_x(x) = p_z(z)|det(\frac{\partial F(x)}{\partial x})| \tag{3.14}$$

$$\ln p_x(x) = \ln p_z(z) + \ln |detJ_F(x)| \tag{3.15}$$

where $J_F(x)$ is the Jacobian of F evaluated at x. Since i-ResNets are guaranteed to be invertible, they can be used to parameterize F in Equation (3.15). For $z = F(x) = (I+G)(x)$, it becomes

$$\ln p_x(x) = \ln p_z(z) + tr(\ln(I + J_G(x))) \tag{3.16}$$

The trace of the matrix logarithm can be expressed as a power series

$$tr(\ln(I + J_G(x))) = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{tr(J_G^k)}{k} \tag{3.17}$$

which converges if $\| J_G \|_2 < 1$.

The log-determinant can therefore be computed using the aforementioned power series with guaranteed convergence due to the Lipschitz constraint.

The log-determinant in Equation (3.16) is estimated using the power-series approximation (Equation (3.17)) with the following algorithm:

Algorithm 2. Forward pass of an invertible ResNets with Lipschitz constraint and log-determinant approximation, SN denotes spectral normalization.

Input: data point x, network F, residual block g, number of power series terms n
for Each residual block **do**
  Lip constraint: $\hat{W}_j := \text{SN}(W_j, x)$ for linear Layer W_j.
  Draw v from $\mathcal{N}(0, I)$
  $w^T := v^T$
  $\ln \det := 0$
  for $k = 1$ **to** n **do**
    $w^T := w^T J_g$ (vector-Jacobian product)
    $\ln \det := \ln \det + (-1)^{k+1} w^T v/k$
  end for
end for

Figure 3.8: Algorithm for log-determinant approximation [2]

Chapter 4

Methodology

4.1 Dataset

The dataset used in this work, Voice-Bank DEMAND dataset [18], is made publicly available by C. Valentini-Botinhao at the Centre for Speech Technology Research (CSTR), School of Informatics, University of Edinburgh. The dataset is freely available for download at `https://datashare.ed.ac.uk/handle/10283/2791`.

The dataset contains clean and corresponding noisy speech audio files of 30 speakers, 14 males and 14 females. A training set containing 28 speakers and a testing set containing 2 speakers were created. In the training set, the utterances were mixed with eight real-world noise samples sourced from the DEMAND database, available here: `http://parole.loria.fr/DEMAND/` and two artificial samples (babble and speech-shaped) at signal-to-noise ratio (SNR) levels of 0, 5, 10, and 15 dB to create 40 different noisy conditions. For the testing set, the utterances were mixed with five different noise samples at SNR values of 2.5, 7.5, 12.5, and 17.5 dB, resulting in a total of 20 different noisy conditions.

The database was created in order to train and test speech enhancement methods. The speech database was obtained from the CSTR VCTK Corpus, which is accessible at `https://doi.org/10.7488/ds/1994`. The speech-shaped and babble noise files that were used in this dataset can be found here: `http://homepages.inf.ed.ac.uk/cvbotinh/se/noises/`.

To enable the application of the PESQ (Perceptual Evaluation of Speech Quality) metric for evaluation purposes, all of the utterances in the dataset were resampled to a sampling rate of 16 kHz.

4.2 Pre- and Post- Processing

Pre-processing refers to the steps taken to prepare raw data for further analysis. It involves cleaning, transforming, and representing data in a format that is suitable for further operations by the proposed model. Post-processing steps are taken on the outputs after applying the SE Model to derive meaningful insights, apply evaluation metric, or present the results in a more understandable form.

The model operates on time-frequency (T-F) domain spectral features. So, the following processing steps are performed on the data for suitable data representation and domain transformations:

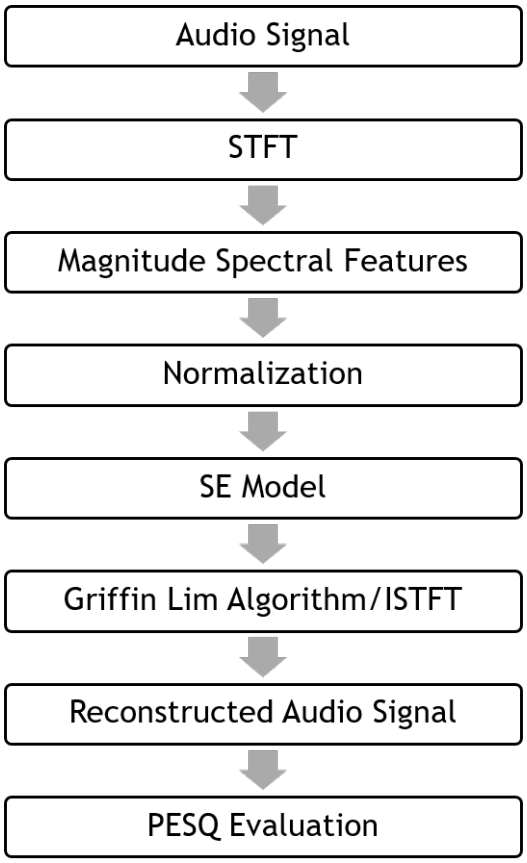

Figure 4.1: Flow Chart

- **Raw data**

  Audio signal waveforms (.wav files) of 2 or more seconds, recorded by a single microphone, are taken and resampled to 16 kHz sampling rate. The low-pass filter width parameter is employed to regulate the filter's width for windowing the interpolation process otherwise the filter would extends infinitely. Increasing the low-pass filter width results in a sharper and more precise filter, but is more computationally expensive.

- **STFT**

  STFT stands for Short-Time Fourier Transform. It is a commonly used signal processing technique that provides a time-frequency representation of a signal by computing the Fourier transform of a windowed section of the signal at successive time frames. It is calculated by dividing the signal into small segments, usually with a fixed duration, and applying the Fourier transform to each segment. This windowed Fourier transform reveals the frequency components present in the signal at that specific time frame. By sliding the window along the signal and repeating the process, a spectrogram or time-frequency representation can be obtained.

  The resampled audio signals of speech dataset are transformed using a short-time Fourier Transform (STFT) with a window size of 510, a hop length of 128, and a Hann window to obtain the complex-valued STFT coefficients matrix.

- **Magnitude Spectral Features**

  The magnitude spectrum features are extracted as a sequence of 256 STFT frames and 256 frequency bins from the complex STFT coefficients, which are further normalized and fed to the model.

- **Normalization**

  The inputs fed to the model is in the form of a single channel 256 X 256 matrix (tensor) normalized with the mean and standard deviation as:

$$\text{output} = \frac{\text{(input - mean)}}{\text{standard deviation}} \tag{4.1}$$

- **SE Model**

  The proposed model performs the speech enhancement task and gives the predicted clean output as magnitude spectral features.

- **Griffin Lim Algorithm**

  The Griffin-Lim algorithm is applied on the output of the model to retreive the audio signals back. It is an iterative algorithm used for phase retrieval in audio signal processing. The goal of the algorithm is to reconstruct the time-domain signal from its magnitude spectrogram, assuming that the phase information is lost or unavailable.

The algorithm involves the following steps:

a. Starting with the magnitude spectrogram of the signal, obtained through STFT, initialize the phase spectrogram randomly or with some initial guess.

b. Reconstruct the time-domain signal by applying the ISTFT to the magnitude spectrogram and the estimated phase spectrogram.

c. Compute the STFT of the reconstructed signal to obtain a new estimate of the phase spectrogram.

d. Update the estimated phase spectrogram by taking the phase of the newly computed spectrogram.

Above three steps are iterated for a fixed number of iterations or until convergence. Once the iterations are complete, the final reconstructed time-domain signal is obtained by applying the ISTFT to the magnitude spectrogram and the last estimated phase spectrogram.

- **PESQ**

  Perceptual Evaluation of Speech Quality (PESQ) metric is applied on the reconstructed audio signal and the target clean signal to evaluate and compare the performance of the proposed model.

4.3 Model

4.3.1 Architecture

The proposed model includes pre-activation ResNets with 13 convolutional bottleneck blocks, where each block comprises three convolutional layers each preceded by ELU non-linearity. The kernel sizes used within these blocks are 3x3, 1x1, and 3x3, respectively. For normalization, ActNorm is applied before each residual block in the invertible models.

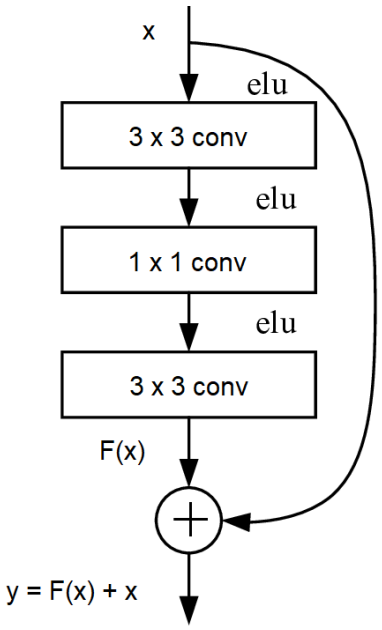

Figure 4.2: Residual block

Each residual block in an i-ResNet is made up of two paths: a forward path and a backward path. The forward path uses skip connections in the same manner as the conventional ResNet to perform the standard convolutional operations.

Let F(x) be the invertible bottleneck block, x be the input to the first layer, then the output of forward path of residual block is given as

$$y = F(x) + x \tag{4.2}$$

The goal of the backward or inverse path is to invert the computations of the forward path in order to recover the original input. The ResNet block is inverted simply by

$$x = y - F(x) \tag{4.3}$$

Figure 4.3 depicts the proposed framework for speech enhancement. The forward path maps the clean speech to noisy speech and the inverse path learns the noisy to clean mapping implicitly during training.

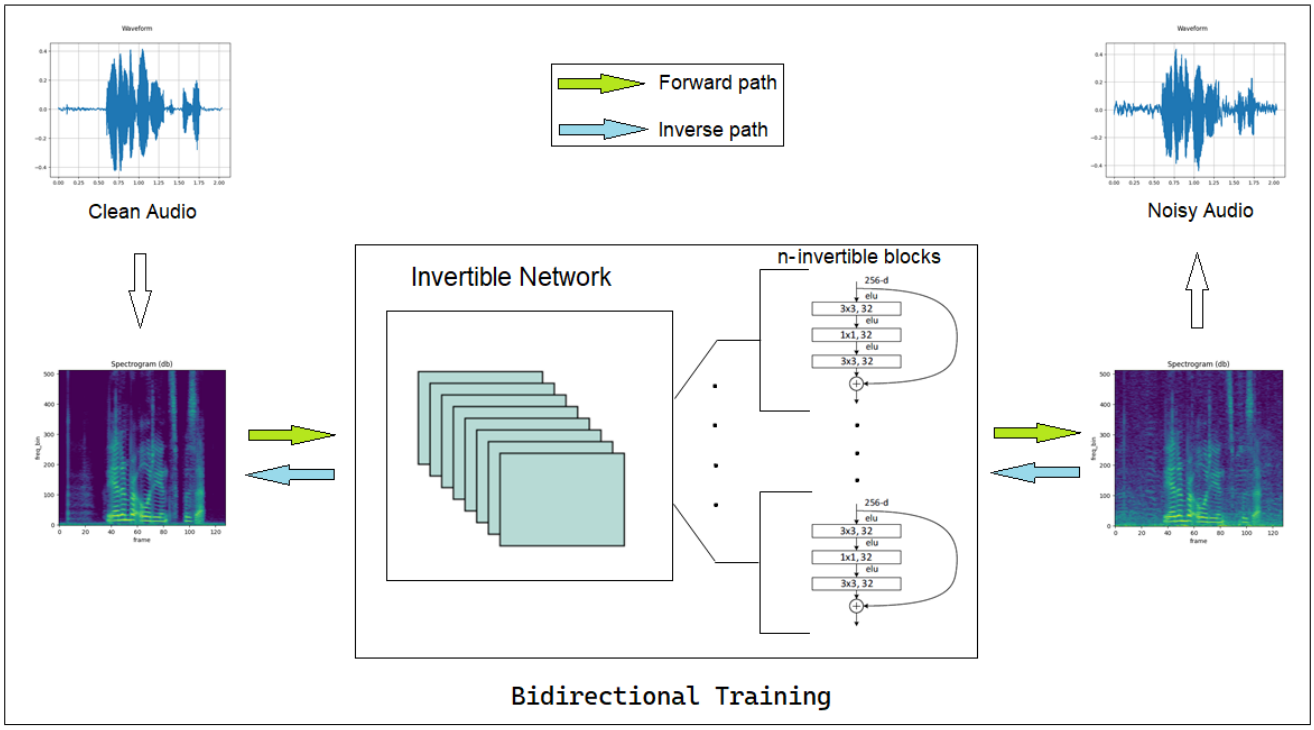

Figure 4.3: Proposed Speech Enhancement Framework

The clean audio input files are transformed to T-F domain using STFT and fed to the i-ResNet after preprocessing. The model transform the data to a predicted output tensor to which we apply Batch-Norm, and a nonlinearity. MSE loss between the predicted and target samples is used for training the model.

The architecture of the generative models closely resembles that of Glow. The model consists of three "scale-blocks" which are groups of i-ResNet blocks. Each scale-block has 16 i-ResNet blocks with 512 filters per convolutional layer. The log-determinant is estimated using the power-series approximation with five terms.

Loss used in training density estimation model is in Bits per dim calculated as:

$$bits/dim = -\log p_x(x)/\log(2) * no.of pixels + 8 \tag{4.4}$$

$$where \log p_x(x) = \log p_z(z) + trace \tag{4.5}$$

Tables 4.1 and 4.2 list the hyper-parameters used in the discriminative and generatice network architectures respectively.

Blocks	13
Channels	32
Lipschitz coefficient	0.7
Learning rate	0.1
Weight decay	5e-4
Non-linearity	elu
Optimizer	momentum sgd
Normalization	actnorm

Table 4.1: Hyper-parameters used in training dicriminative model

Blocks	16
Channels	512
Lipschitz coefficient	0.7
Learning rate	0.003
Non-linearity	elu
Optimizer	adamax
Power series term	5

Table 4.2: Hyper-parameters used in training generative model

4.3.2 Normalization

Normalization in neural networks refers to the process of transforming the inputs or activations of a neural network layer to a standardized scale or distribution so as to make the network more robust, improve convergence during training, and facilitate better generalization. It can help accelerate training, reduce overfitting, and improve the overall performance of neural networks.

- **Activation Normalization**

  ActNorm, short for Activation Normalization, is a technique used in deep learning models to normalize the activations of neural network layers. The goal of ActNorm is to ensure that the activations of the network layers are centered around zero with unit variance.

  The ActNorm follows two steps:

  1. **Initialization:** The network layer is initialized such that its activations have zero mean and unit variance.
  2. **Activation Rescaling:** During the forward pass of the network, the activations are rescaled to maintain zero mean and unit variance. This rescaling is applied element-wise to each activation independently.

- **Batch Normalization**

  Batch normalization (BatchNorm) aims to address the internal covariate shift problem, which refers to the phenomenon where the distribution of inputs to each layer of a network changes during training, making it challenging for the network to learn effectively.

  The basic idea behind batch normalization is to normalize the inputs to a layer by subtracting the batch mean and dividing by the batch standard deviation. This process is applied independently to each feature dimension of the layer's input. The normalization is performed over a mini-batch of training examples, hence the name "batch" normalization.

4.3.3 Choice of Activation Function

The choice of non-linearity depends on the specific problem and architecture. In the proposed model second derivatives of the network output are computed to differentiate the log-determinant estimator. These values are not determined in some locations if a non-linearity with discontinuous derivatives (such as ReLU) is used. Activation functions with continuous derivatives, such as ELU or softplus are desirable, to ensure that the quantities necessary for optimisation always exist.

- **ReLU**

  Rectified Linear Unit, sometimes known as ReLU, is a non-linear activation function commonly used in artificial neural networks. The function and its derivative both are monotonic. It has a range from 0 to infinity and is defined as:

$$f(x) = \begin{cases} 0, & \text{for } x < 0. \\ x, & \text{for } x \geq 0. \end{cases} \tag{4.6}$$

  ReLU accepts the value x as an input and returns the maximum of that value and zero. It returns the input value itself if the input is positive, and returns 0 if the input is negative.

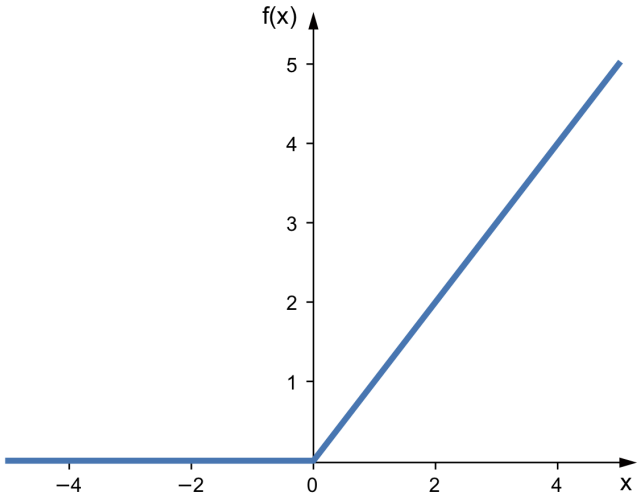

Figure 4.4: ReLU Activation Function

- **ELU**

  An Exponential Linear Unit activation function takes an input x and performs identity operation if it is positive, and an exponential non-linearity if the input is negative. It is defined as:

$$f(x) = \begin{cases} \alpha(e^x - 1), & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases} \tag{4.7}$$

ELU has the property of being differentiable everywhere, which is important for efficient gradient-based optimization algorithms.

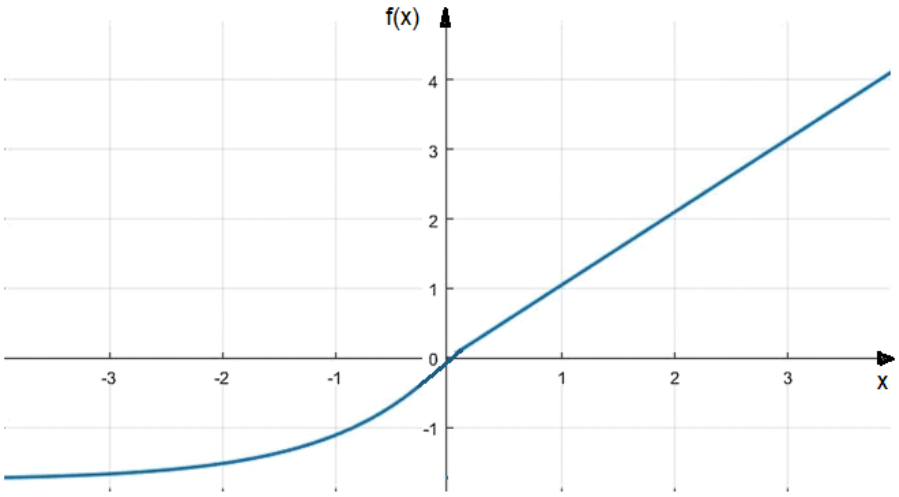

Figure 4.5: ELU Activation Function

4.3.4 Choice of Optimizer

An optimizer is an algorithm used to adjust the parameters of the model during the training process in order to minimize the loss function and improve the model's performance. The goal of an optimizer is to find the set of parameter values that result in the lowest possible value of the loss function. The proposed model uses the following optimizers after empirical evaluations:

- **Adamax**

  Adamax is a variant of the Adam optimizer applied to models with sparse gradients or large parameter spaces. It uses momentum to keep track of the exponentially decaying average of past gradients and adapts the learning rate for each parameter based on the infinity norm of past gradients. By using the infinity norm instead of the L2 norm as in Adam, Adamax reduces the effect of scaling on the update step computation, allowing for more stable training.

- **Stochastic Gradient Descent with momentum**

  This optimizer builds upon the basic SGD algorithm by incorporating a momentum term that helps to dampen oscillations and accelerate gradient updates in relevant directions. The momentum term enables the optimizer to increase velocity in the direction of consistent gradients and reduce the updates in directions where the gradients change rapidly or oscillate. This helps to accelerate convergence and smooth out the optimization path, leading to faster training and potentially better generalization.

4.4 Evaluation Metric

4.4.1 MSE Loss

It is a metric that quantifies the mean squared error, also known as the squared L2 norm, between each element in the predicted output $\hat{y}$ and target y. MSE loss between the predicted and target noisy speech is used during the training of the proposed model.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.8}$$

The MSE loss function is often used as the optimization objective in regression problems when the goal is to minimize the average squared difference between predictions and actual values. It is differentiable and convex, making it well-suited for optimization algorithms such as gradient descent.

4.4.2 PESQ

PESQ stands for Perceptual Evaluation of Speech Quality. It is a widely used objective method for measuring the perceived quality of speech signals. PESQ is a full reference algorithm which assesses the degradation of speech quality caused by various factors such as noise, distortion, and network transmission issues.

PESQ is often used in telecommunications and audio processing industries to evaluate the performance of speech codecs, communication networks, and other speech processing algorithms [14]. It provides a numerical score, called the PESQ score, which indicates the perceived quality of the speech signal. The score ranges from -0.5 to 4.5, higher the PESQ value the better is perceived quality.

PESQ works by comparing the reference (original) speech signal with the degraded speech signal using a model of human hearing. It takes into account factors like speech distortion, delay, and noise to calculate the score.

Chapter 5

Results and Discussion

The invertibility of the proposed i-ResNet models is verified on Voice-Bank DEMAND dataset. It's predictive ability on regression task is also investigated. Further, utilization of i-ResNet as generative model is studied.

The models are trained on NVIDIA Tesla T4 using Google Colab in PyTorch [15]. MSE and bits per dim loss vlaues are used for training the dicriminative and generative models respectively. The proposed model performs enhancement in the time-frequency domain.

The standardized evaluation metric for performance comparison, perceptual evaluation of speech quality (PESQ) is reported.

Initial experiments are performed on 3-channel RGB spectrogram images, then single channel magnitude spectral coefficients of STFT matrix are used to facilitate application of PESQ algorithm.

5.1 Training Results

In this work, the proposed iResNet model is trained for 100 epochs with momentum SGD and a weight decay of 5e-4 on approximately one hour of Voice-Bank DEMAND dataset. The learning rate is set to 0.1 and decayed by a factor of 0.2 after 60 epochs.

The training set consists of approximately 40 minutes of clean and corresponding noisy speech data, 1200 utterances of 2 seconds each, sampled at 16kHz.

The input to our first residual block is a tensor of size 256x256x1 obtained from the STFT matrix. For data-augmentation, random horizontal flips are applied during training.

The inputs are normalized by subtracting the mean and dividing by the standard deviation of the training set.

The mean squared error of the network during training is shown in the following figure to illustrate that the output of the network converges to the desired target and learning is done successfully.

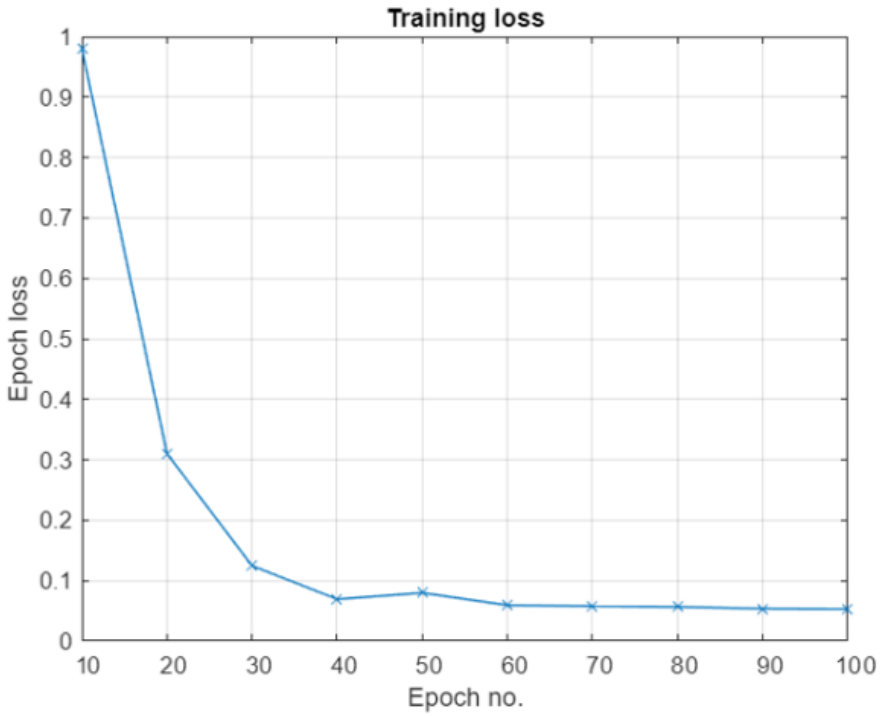

Figure 5.1: MSE loss v/s No. of epochs

The training versus validation loss of a subset of dataset is also shown in the figure:

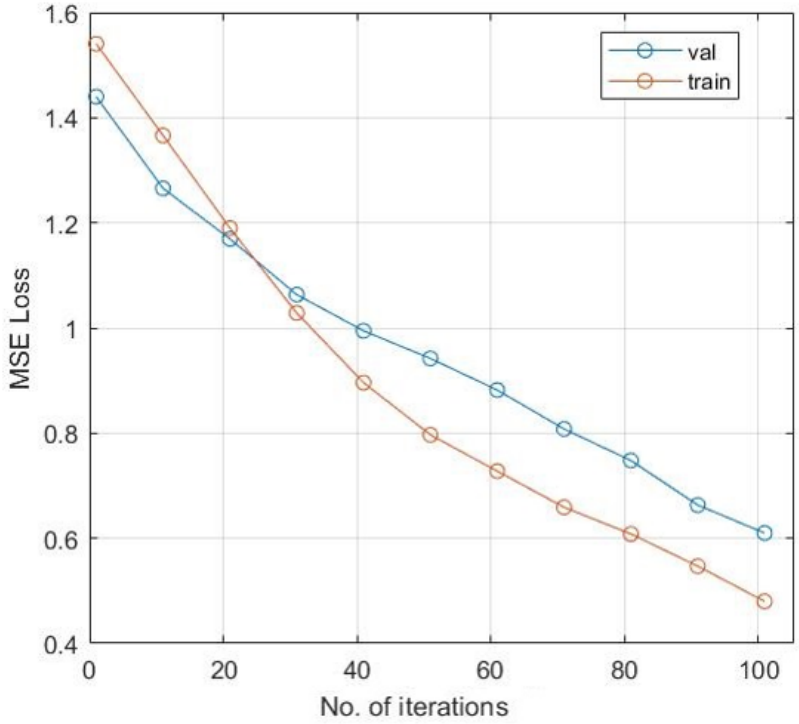

Figure 5.2: Training v/s Validation Loss

The density estimation model is trained for 50 epochs using the Adamax optimizer with a learning rate of 0.003 as per the log-determinant approximation (Algorithm 2 described in Chapter 3). The log determinant is estimated using the power-series approximation with five terms for the model during training.

The bits per dimension error of the network during training is shown in the following figure, lower values indicate better system performance.

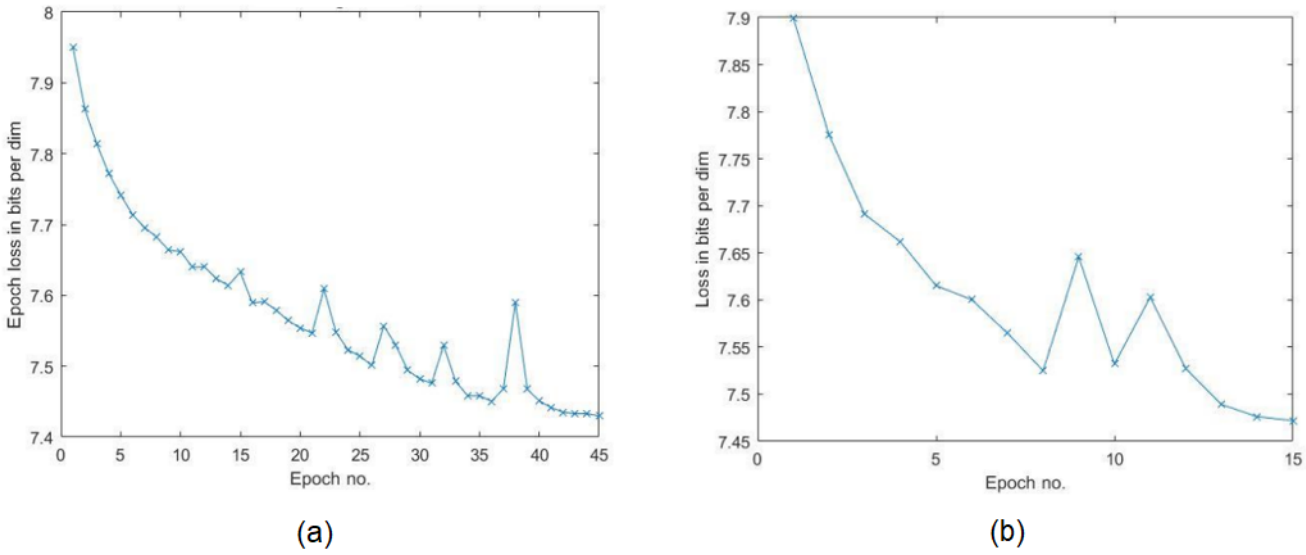

Figure 5.3: Bits per dim v/s No. of epochs

From the Figure 5.3 (a), it can be observed that the loss is converging slowly so the model is optimized by tweaking the hyper-parameters to obtain the same loss at the end of 1/3rd the number of epochs as shown in Figure 5.3 (b).

While the proposed dE model did not perform as well, it is noteworthy that ResNets, with very little alterations, can build a generative model. The use of an unbiased log-determinant estimator can help improve the performance [2]

5.2 Testing Results

The test set consists of approximately 20 minutes of unseen noisy speech data, around 500 utterances of 2 seconds each, sampled at 16kHz. The best model is saved based on the minimum loss after training, and tested on the test dataset.

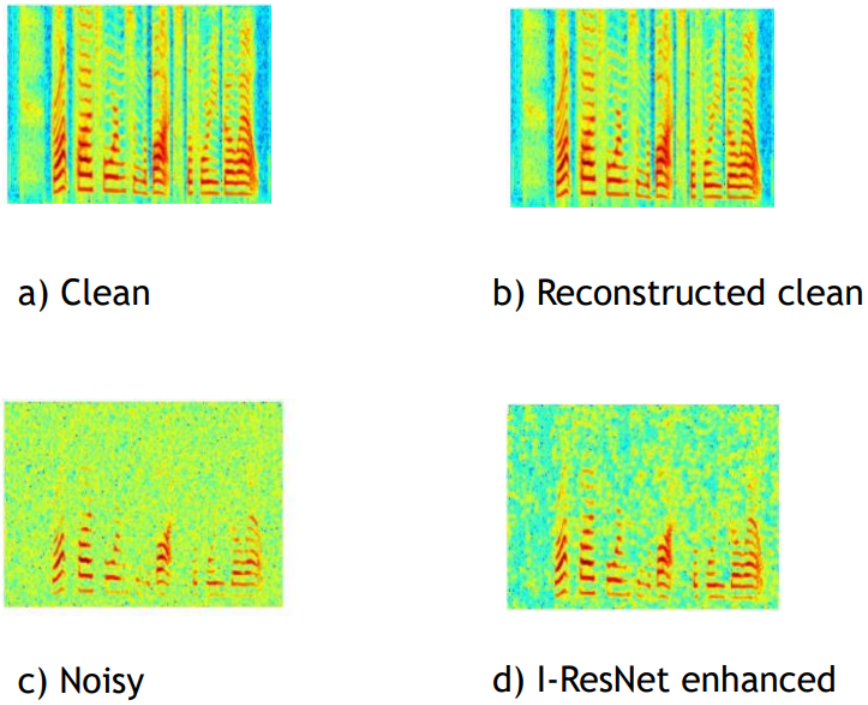

Figure 5.4: Spectrograms of an utterance sample

Figure 5.4 shows the spectrograms of one of the utterance sample with the original clean speech spectrogram in (a) and the corresponding reconstructed spectrogram from the inverse path in (b), noisy speech spectrogram and the enhanced clean spectrogram are displayed in (c) and (d) respectively. It can be noted from the Figure 5.4 (b) that the i-ResNet is invertible and gives perfect reconstruction of the input samples.

5.2.1 PESQ evaluation

The PESQ metric is applied on the predicted output i.e., enhanced speech signals from the noisy test set in mismatched noisy conditions. The PESQ score ranges from -0.5 to 4.5, with higher values indicating better perceived quality.

Evaluation results and comparison of iResNet model based on PESQ values with others on Voice-Bank corpus is presented in the following table:

Model	**PESQ**
SEGAN	2.16
i-ResNet	1.71
DiffuSE (Base)	1.97

Table 5.1: PESQ results among SEGAN, i-ResNet and DiffuSE on the test set

The PESQ values for SEGAN and DiffuSE (Base) have been quoted from [12] in the above table. It can be observed from the Table 5.1 that though the proposed model is yet to achieve the best overall results, it is comparable with other SOTA methods of speech enhancement and can be improved upon.

Chapter 6

Conclusions and Future Works

6.1 Conclusions

Modern speech enhancement techniques often combine multiple approaches and incorporate deep learning methods, utilizing their ability to learn complex mappings between noisy and clean speech signals.

This thesis examines the performance of i-ResNet model, which combines the principles of invertible mappings with residual connections, for speech enhancement task. The proposed model is the first application of iResNet for a regression problem, i.e., speech enhancement. It also presents an approach to close the gap between generative and discriminative models for regression by utilizing one unified architecture for both.

Evaluations on Voice-Bank DEMAND dataset show that the performance is comparable to other methods and can be further improved by advanced training losses and data augmentation. Many facets of the model remain unexplored, and many early design choices remain unaltered [2]. The loss function, and waveform synthesizer based on Griffin-Lim both have room for improvement.

6.2 Future Works

The scope for the future work can be summarized as follows:

- Phase information of the noisy signal can be retained and used in ISTFT evaluations rather than using the Griffin Lim algorithm.
- Instead of using the magnitude spectral features, complex-valued STFT inputs for combined magnitude and phase enhancement can be used so that phase information is also utilized in enhancement.
- In order to improve the proposed model better loss functions more relevant to human auditory perception can be explored.
- Other relevant evaluation metrics can be employed as well as subjective listening experiments can be carried out for qualitative evaluation.

Bibliography

[1] L. Ardizzone et al. “Analyzing Inverse Problems with Invertible Neural Networks”. In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=rJed6j0cKX.

[2] J. Behrmann et al. “Invertible Residual Networks”. In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 573–582. URL: https://proceedings.mlr.press/v97/behrmann19a.html.

[3] S. Boll. “Suppression of acoustic noise in speech using spectral subtraction”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 27.2 (1979), pp. 113–120. DOI: 10.1109/TASSP.1979.1163209.

[4] Y. Ephraim and D. Malah. “Speech enhancement using a minimum mean-square error log-spectral amplitude estimator”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 33.2 (1985), pp. 443–445. DOI: 10.1109/TASSP.1985.1164550.

[5] Y. Ephraim and H.L. Van Trees. “A signal subspace approach for speech enhancement”. In: *IEEE Transactions on Speech and Audio Processing* 3.4 (1995), pp. 251–266. DOI: 10.1109/89.397090.

[6] P. L. Fernández-Cabán, F. J. Masters, and B. M. Phillips. “Predicting Roof Pressures on a Low-Rise Structure From Freestream Turbulence Using Artificial Neural Networks”. In: *Frontiers in Built Environment* 4 (2018). ISSN: 2297-3362. DOI: 10.3389/fbuil.2018.00068. URL: https://www.frontiersin.org/articles/10.3389/fbuil.2018.00068.

[7] D. Gutermuth. “Picard’s Existence and Uniqueness Theorem”. In: 2003. URL: https://ptolemy.berkeley.edu/projects/embedded/eecsx44/lectures/Spring2013/Picard.pdf.

[8] K. He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[9] G. Kim et al. “An algorithm that improves speech intelligibility in noise for normal-hearing listeners”. In: *The Journal of the Acoustical Society of America* 126 (Oct. 2009), pp. 1486–94. DOI: 10.1121/1.3184603.

[10] J.S. Lim and A.V. Oppenheim. “Enhancement and bandwidth compression of noisy speech”. In: *Proceedings of the IEEE* 67.12 (1979), pp. 1586–1604. DOI: 10.1109/PROC.1979.11540.

[11] P. C. Loizou. *Speech Enhancement: Theory and Practice*. Jan. 2007. ISBN: 9780429096181. DOI: 10.1201/b14529.

[12] Y.-J. Lu, Y. Tsao, and S. Watanabe. “A Study on Speech Enhancement Based on Diffusion Probabilistic Model”. In: *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. 2021, pp. 659–666.

[13] S. Pascual, A. Bonafonte, and J. Serrà. “SEGAN: Speech Enhancement Generative Adversarial Network”. In: *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*. ISCA, 2017, pp. 3642–3646.

[14] *Perceptual evaluation of speech quality (PESQ) and objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs,* 2000.

[15] *PyTorch Documentation*. URL: https://pytorch.org/docs/stable/index.html.

[16] S. Tamura. “An analysis of a noise reduction neural network”. In: *International Conference on Acoustics, Speech, and Signal Processing,* 1989, 2001–2004 vol.3. DOI: 10.1109/ICASSP.1989.266851.

[17] S. Tamura and A. Waibel. “Noise reduction using connectionist models”. In: *ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing*. 1988, 553–556 vol.1. DOI: 10.1109/ICASSP.1988.196643.

[18] C. Valentini-Botinhao. *Noisy speech database for training speech enhancement algorithms and TTS models, 2016 [sound]*. University of Edinburgh. School of Informatics. Centre for Speech Technology Research (CSTR), 2017. URL: https://doi.org/10.7488/ds/2117.

[19] Y. Xu et al. “An Experimental Study on Speech Enhancement Based on Deep Neural Networks”. In: *IEEE Signal Processing Letters* 21.1 (2014), pp. 65–68. DOI: 10.1109/LSP.2013.2291240.

[20] H. Zhao et al. “Convolutional-Recurrent Neural Networks for Speech Enhancement”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018, pp. 2401–2405. DOI: 10.1109/ICASSP.2018.8462155.