Université du Québec
Institut national de la recherche scientifique
Centre Énergie Matériaux Télécommunications

Revisiting Automatic Speech Recognition Systems for Augmentation Capacity

By
Anwar Tantawy

A thesis submitted in fulfillment of the requirements for the degree of
Doctorate of Sciences, Ph.D
in Telecommunications

Evaluation Committee

Internal evaluator and Committee president	Prof. Amar Mitiche INRS-EMT at Montreal
External evaluator 1	Prof. Changxue Ma GC Resound company, Illinois, US
External evaluator 2	Prof. T. Nagarajan Shiv Nadar University, Chennai, India
Research Director	Prof. Douglas O'Shaughnessy INRS-EMT at Montreal

Dedicated To

My beloved wife Maram.

My wonderful mother and father, Maram and Dr. Hasan

My beautiful children who are the joy of my life; Adam, Sally, Laura, and Eva

ACKNOWLEDGMENTS

No matter how hard I try, I can never thank professor Douglas O'Shaughnessy enough for his continuous support of my work. He is a true educator; a giant in his field, a father figure, and always a friend. I am truly blessed to have attended a great institution such as INRS under his wing. I would also like to thank the staff at INRS who are always very professional, kind, and caring. During my time at INRS-EMT, I believe that I have retained the proper knowledge and insight to continue my journey in the field of Automatic Seech Recognition.

I would also like to acknowledge the president of the committee and my internal evaluator, professor Amar Mitiche, as well as my external evaluators; professor Changxue Ma (GC Resound company, Illinois, US) and professor T. Nagarajan (Shiv Nadar University, Chennai, India). It is an honor to defend my thesis in front of greats as you.

I am grateful for my beautiful lovely wife and best friend, Maram Almajali for her love and continuous support. I am so lucky to have you in my life. My son Adam has been an amazing loving and caring friend throughout this journey. My daughter Sally is my forever princess. My two daughters Laura and Eva were born during this thesis work, and I am truly blessed to have them running around my desk every day.

My dear mother and father; Maram Bilbaisi, and Dr. Hasan Tantawy have always been my umbrella of safety and comfort. My mother's prayers always reach me no matter where I am. My dear in-laws have supported us as a family every day without question. I am blessed to have the most wonderful mother-in-law and father-in-law on this planet; Ghada Almajali, and Dr. Fayez Almajali. Finally, with me being the eldest of eight, I would like to thank all of my siblings who I am so very proud of all of their accomplishments; Dr. M. Noor Tantawy, Dr. Diya Tantawy, Dr. Lara Tantawy, Dr. Khalid Tantawy, Eng. Sa'ad Tantawy, Eng. Omar Tantawy, and IT-instructor Abdelrahman Tantawy. I love you all my family so dearly. I am nothing without you.

Finally, I have to thank and acknowledged all the support I have received from my friends and neighbors. You all have been amazing and wonderful. You have been more like family to me. God bless you all!

This Page Left Blank Intentionally

ABSTRACT

Automatic Speech Recognition (ASR) applications have increased greatly during the last decade due to the emergence of new devices and home automation hardware that can benefit greatly from allowing users to interact hands free, such as smart watches, earbuds, portable translators, and home assistants. ASR implementation for these applications inevitably suffers from performance degradation in real life scenarios. Most ASR systems expect the working environment to be like the training environment, which is often not the case, especially for new applications with limited data availability. This study is concerned with experimentally showing the effect of variations in the environment on different ASR models and the capacity of different models to improve performance when provided with training data like the testing environment.

Taking a certain type of variability into account takes place by modifying or adapting one of the ASR system components, thus alleviating the effect of variability in real-life scenarios. However, this nominal approach does not account for all possible variabilities simultaneously, but on the contrary might result in deterioration in performance against other types of changes in the testing environment.

Most of the recent successes in ASR are mainly dependent on the abundance of data in a certain domain along with the increased capacity of the learning models. The performance of ASR then decreases with the decrease of either the amount of data or model capacity. Hence, this work proposes different data augmentation techniques and focuses on the capacity of the different models to improve with different types of augmented data.

Key words **– Acoustic Modelling, Data Augmentation, Recurrent Autoencoder, Neural Style Transfer**.

RÉSUMÉ

Les applications de reconnaissance automatique de la parole (ASR) ont considérablement augmenté au cours de la dernière décennie en raison de l'émergence de nouveaux appareils et de matériel domotique qui peuvent grandement bénéficier de la possibilité pour les utilisateurs d'interagir les mains libres, tels que les montres intelligentes, les écouteurs, les traducteurs portables et les assistants domestiques. La mise en œuvre d'ASR pour ces applications souffre inévitablement d'une dégradation des performances dans les scénarios réels. La plupart des systèmes ASR s'attendent à ce que l'environnement de travail ressemble à l'environnement de formation, ce qui n'est souvent pas le cas, en particulier pour les nouvelles applications avec une disponibilité limitée des données. Cette étude vise à montrer expérimentalement l'effet des variations de l'environnement sur différents modèles ASR et la capacité de différents modèles à améliorer les performances lorsqu'ils sont fournis avec des données d'entraînement comme l'environnement de test.

La prise en compte d'un certain type de variabilité se fait en modifiant ou en adaptant l'un des composants du système ASR, atténuant ainsi l'effet de la variabilité dans des scénarios réels. Cependant, cette approche nominale ne tient pas compte de toutes les variabilités possibles simultanément, mais au contraire pourrait entraîner une détérioration des performances par rapport à d'autres types de changements dans l'environnement de test.

La plupart des succès récents en ASR dépendent principalement de l'abondance de données dans un certain domaine ainsi que de la capacité accrue des modèles d'apprentissage. Les performances de l'ASR diminuent alors avec la diminution de la quantité de données ou de la capacité du modèle. Par conséquent, ce travail propose différentes techniques d'augmentation de données et se concentre sur la capacité des différents modèles à s'améliorer avec différents types de données augmentées.

Mots clés **– Modélisation acoustique, augmentation de données, auto-encodeur récurrent, transfert de style neuronal.**

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

AM	*Acoustic Model*
ANN	*Artificial Neural Network*
ASR	*Automatic Speech Recognition*
CH	*CallHome*
CNN	*Convolutional Neural Network*
CRF	*Conditional Random Fields*
DA	*Data Augmentation*
DFT	*Discrete Fourier Transform*
DNN	*Deep Neural Network*
DTW	*Dynamic Time Warping*
EM	*Expectation-Maximization*
GMM	*Gaussian Mixture Model*
HMM	*Hidden Markov Model*
LAS	*Listen, Attend, and Spell network*
LDA	*Linear Discriminant Analysis*
LM	*Language Model*
LPCs	*Linear Predictive Coefficients*
LPF	*Low Pass Filter*
LSTM	*Long Short-Term Memory*
MAP	*Maximum A Posteriori*
MFCCs	*Mel-Frequency Cepstrum Coefficients*
MLLT	*Maximum Likelihood Linear Transform*
MMD	*Maximum Mean Discrepancy*
NSP	*Nationwide Speech Project*
NST	*Neural Style Transfer*
PER	*Phone Error Rate*
RAE	*Recurrent Autoencoder*
RBM	*Restricted Boltzmann Machine*
ReLU	*Rectifier Linear Unit*
RGB	*Red, Green, Blue*
RNN	*Recurrent Neural Network*
RVAE	*Recurrent Variational Autoencoder*
S2S	*Sequence-to-Sequence*
SAT	*Speaker Adaptive Training*
SGD	*Stochastic gradient descent*
SGMM	*Subspace Gaussian Mixture Model*
SNR	*Signal to Noise Ratio*
SSA	*Speech Style Augmentation*
SWB	*Switchboard*
TTS	*Text-To-Speech*
VAE	*Variational Autoencoder*
VC	*Voice Conversion*
WER	*Word Error Rate*

Chapter 1

Introduction

The popularity of Automatic Speech Recognition (ASR) is without a doubt soaring, solidifying its significance within the technological sphere. ASR has become part of millions of people's daily routine whether be it vehicles, appliances, smart phones, Amazon Echo-Alexa, Siri, Cortana, Google, and much more [1]. Speech recognition has enabled users to connect with these devices and gadgets more fluidly, potentially revolutionizing the human-machine interaction environment. The purpose of a speech recognition system is to properly transform an audio waveform (speech signal) into words via a computer interface, regardless of the speaker or ambient variables. To put it another way, ASR is a system that accepts a speech signal as an input and outputs words that match the input speech signal. Converting a voice signal into words is a difficult undertaking since speech signals are naturally complex. There are several variations: physiological, environmental, linguistic, and so on.

1.1 ASR Challenges

A speech signal is simply decomposed into the source and filter, where the source being the human vocal folds in voiced speech and the filter being one's vocal tract and articulators. How people tend to pronounce words is referred to as accents. Accents are a part of a dialect which also encompasses grammar and vocabulary. It is challenging enough to have an ASR system recognize speech with different accents, but even more so when different dialects are thrown in. With English being the international language of the world, it is worth mentioning how American and British dialects can differ distinctively in accents, grammar, and vocabulary.

There are roughly 30 major dialects in America, while there are approximately 40 dialects in the UK alone. These dialects not only may differ in accents, but in word-spelling and word structure as well [2]. For example, collective nouns in American English are considered singular (e.g., The team is celebrating), whereas in British English, collective nouns can be either singular or plural (e.g., The team are celebrating). Other examples are how some vocabularies are used in very different ways between American and British English in respect (e.g., Vacation vs. Holiday, Cookie vs. Biscuit, Pants vs. Trousers), or "I would like to take a walk in the park" vs. "I fancy a walk in the park".

Despite all the breakthroughs ASR has achieved during the past four decades, it is still bounded by many limitations that degrade its performance. A significant constraint on ASR performance is ambient noise and reverberation, which distort speech when captured by microphones [1]. Noises, including reverberation, can be classified as stationary where it is constant in time, or non-stationary as in varying with time like transient sound events, interfering speakers, and music. Short-term stationary additive noise can be effectively dealt with using typical, unsupervised noise reduction signal processing techniques as long as reliable detection of instants of the target (speech) signal is achievable. Detecting and mitigating the impacts of non-stationary ambient noise, competing non-stationary sound sources, or highly reverberant situations, on the other hand, remains extremely difficult in reality.

Reverberation is simply the persistence or lagging of sound after the source of that sound is gone, which amounts to room distortion. Figure 1.1 illustrates the time delay it takes for a source sound to decay by 60 dB due to reverberation. Reverberation can be thought of as the convolution of a speech signal and the room impulse response [3]. It smears the signal in time where the signal is convolved with different room impulse responses [4]. One important thing to recognize is that for frame-based algorithms, reverberation not only smears energy of each frame, but across multiple frames that follow.

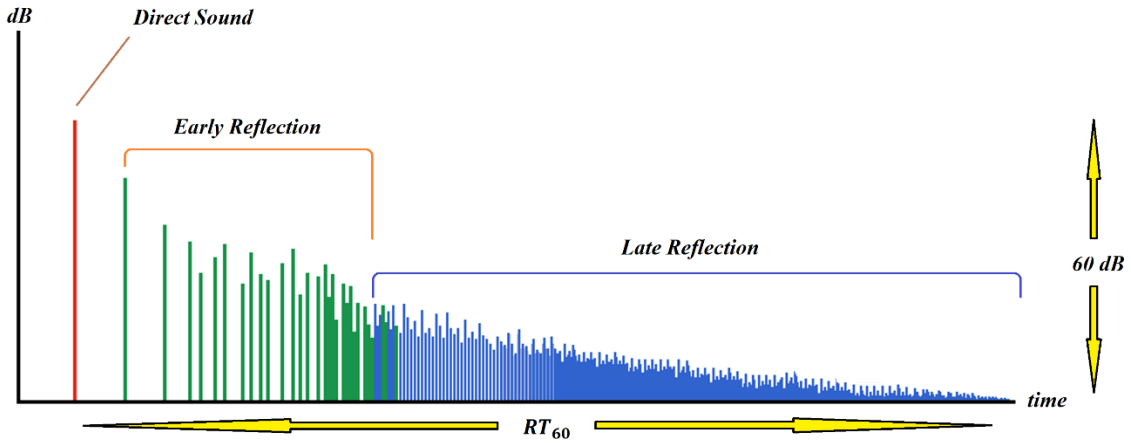


Figure 1.1: Reverberation

1.2 Synthesis for Augmentation

Speech synthesis may be the critical component of low-resource speech recognition. Speech synthesis techniques have made significant progress toward the aim of producing human-like speech [5]. In recent years, considerable attention has been paid to the use of synthesized speech as a kind of data

augmentation in order to reduce the resources required to develop a speech recognizer [5]. Synthesis is an approach to generate speech from text. That makes it a bit of a challenging task in general, especially when trying to use it for enriching speech corpora. That is why another closely related field of study that generates speech data, but from other speech rather than from text can bridge the required gaps. Human speech is defined by four key characteristics: timbre, content, pitch, and rhythm. Timbre is related to the speaker's vocal tract shape and spectral characteristics, content conveys the speech's textual and contextual information, and pitch and rhythm indicate the prosody of the speech. The bulk of voice conversion (VC) techniques concentrate on timbre conversion, altering the source utterance frame by frame while retaining the original speech's prosody [6]. The use of voice conversion (VC) to augment sparse training data can improve speech recognition systems in low-resource languages. Voice conversion is useful for a variety of purposes, including privacy protection, content development for the entertainment sector, and tailored speech synthesis.

To develop a realistic speaking utterance, it is critical to produce natural prosody, which comprises tone, intonation, emphasis, and rhythm. Prosody is defined in acoustics and speech signal processing as the composition of each phoneme’s duration in an utterance, the pitch contour over the whole phrase, and the speaker’s way of changing speech duration and other controllable speech generation functionalities, such as amplitude.

Each of these elements can be examined and synthesized independently or in conjunction with the others, depending on the modelling techniques used. Prosody is influenced by a variety of elements, including the speaker's identity, mood, the aim of the speech, the speech’s form, the semantic or syntactic relationships between lexical words, and the speaker's emphasis. Styles are shared prosodic qualities of an utterance or a group of utterances that are influenced by common elements in speech. For instance, we associate a style with a certain speaker identity based on some common prosodic characteristics seen in the speaker's utterances [7]. Speaking style stands for how speech is expressed, such as in reading, chat, story-telling style, or in some emotional state such as happy and sad.

End-to-end neural Text-To-Speech (TTS) has recently proven that it can synthesize very natural, human-like speech within the constraints imposed by the training data in target styles and speakers. Prosody, mood, and speaker identity can all be modelled and controlled. Single-speaker, multi-style synthesis is becoming more popular, but it is still in its infancy due to expense. Most neural TTS systems now use a corpus of a single expressive style to model them. They all model speech styles into a single style representation, which contains far too much interfering information to be robust and interpretable, and

lacks the capacity to manage a single speech characteristic separately. When transferring styles, one must consider to transfer all styles, whether requested or not, and they may or may not fit the contexts, limiting generalizations [8].

Although the accuracy of ASR systems has improved significantly in recent years, individuals still prefer to interact with computers through a keyboard or touchscreen. The main argument is that, given the low accuracy of ASR, typing is more convenient than dictation. Furthermore, the current boost in performance is restricted to certain circumstances. To have a smooth connection with gadgets and broad usage of ASR, the accuracy of ASR under all scenarios must be improved. Speaker mismatch, channel mismatch, and noise are the three most common differences.

1.3 Thesis Organization

This thesis is organized as follows: This chapter gives a brief overview of how much advancements ASR systems have achieved, yet how much challenges it still faces. It also gives a brief insight of synthesis for data augmentation where training data is synthetically enriched. Chapter 2 entails the Problem-Statement, which presents the scope of the thesis, as well as the research objectives, research questions, and research hypothesis. Chapter 3 provides background and related works in ASR enhancement and data augmentation. Chapter 4 discusses literature attempts in speech enhancement. Chapter 5 discusses variations in speech and presents different datasets used in this study, while Chapter 6 studies the model capacities within environmental variations. Chapters 7, 8, and 9 discuss three newly proposed data augmentation techniques that tackle speaker variability. The proposals are Neural Style Transfer, WaveNet Modelling, and Recurrent Autoencoders. Chapter 10 then concludes the findings of this thesis. Appendix-A demonstrates the challenges we faced during the experimentation phase, including many failed experiments. Appendix-B examines in more detail the structure of the Kaldi toolkit and a general idea of how it functions.

Chapter 2

Problem Statement

2.1 Scope

While the demand for Automatic Speech Recognition (ASR) systems increased in various automation applications such as cars, smart watches, mobile phones, smart homes, dictation, and translation, these applications inevitably suffer from degradation in real life scenarios. Most ASR systems expect that the working environments are similar to the training environment, which hardly is the case. Many sources of variation in speech signals and the background cannot always be anticipated.

The scope of this thesis is to show how can the commonly used state-of-the art ASR systems, namely the hybrid Hidden Markov Model – Deep Neural Network (HMM-DNN) [9], Subspace Gaussian Mixture Model (SGMM) [10], and Sequence-to-Sequence (end-to-end) Recurrent Neural Networks [11], [12] handle variations in a working environment. A dataset enriching approach is also proposed by introducing synthetic variabilities to improve the performance of ASR systems when working in a real setting.

2.2 Research Objectives

This thesis is concerned with finding solutions to the environmental variations between the training and working settings of ASR systems. A core problem with machine learning training algorithms is that they do not generalize well over data they trained on. The main goal is to identify the generalization power of different state-of-the art ASR systems, and their capacity to learn from small portions of distortions introduced in the training set. Moreover, a data augmentation procedure which introduces synthetic common variabilities to the data is proposed for improved performance. Detailed research objectives are:

1. Testing the robustness of various state-of-the-art models and features against variations between training and testing environments.
2. Identifying the capacity of these models and features to learn from data containing multiple sources of interference.
3. Identifying improvements in performance when multiple synthetic sources of variability are introduced.

2.3 Research Questions and Hypothesis

2.3.1 Research Questions:

1. How robust are the modern ASR systems mentioned previously, to variations in the nature of the speaker and background noise of speech?
2. How fast can ASR system performance improve when introduced with small portions of speech variations?
3. How can variations be synthesized to enrich training datasets?
4. How much improvement can be achieved when the training data is synthetically enriched?

2.3.2 Research Hypothesis:

1. The ASR systems' capacity to improve in performance when introduced to different types of variations in the training phase is not constant and depends on both the model and the type of variation.
2. The generation of synthetic data can add information to the training data through either physical distortion or learned distortions. ASR systems can improve in performance when introduced with synthetically generated variations for a given ASR model.

Chapter 3

Background

3.1 Analysis of Speech (Front End)

Speech is generated via a time-varying vocal tract system that results in dynamic or time-varying speech signals. While the speaker has control over many components of speech production, including volume, voicing, fundamental frequency, and vocal tract structure, much speech variation is random, for example, vocal fold vibration is not entirely periodic [13].

The purpose of speech analysis is to extract features from the speech signal by transforming it into another signal or a set of signals. The acoustic model is used to approximate the acoustic aspects of speech; "Temporal" by directly operating on the speech waveform or "Spectral" in the frequency domain after a spectral transformation. In Time-domain analysis, there is an emphasis on short time windowing since speech is highly non-stationary. During Speech processing, it is commonly assumed that speech is quasi-stationary (time invariant), thus a good compromise would be an analysis window of 20 – 30 ms length with 10 ms update [14].

The choice of the Low Pass Filter (LPF) window length brings with it a trade-off between time resolution vs. frequency resolution. A long-time window results in high frequency resolution, but of course poor time resolution, and vice versa. A commonly used window is the Hamming window. It is a good LPF and a smooth window although time-compression means frequency-expansion (wider band), resulting in a higher Nyquist sampling rate to avoid overlapping in the frequency domain. In other words, time-compression requires higher sampling frequency in the time-domain to capture the compressed changes without loss of information. In the Frequency domain, the Short Time Fourier Transform is applied where the speech signal is windowed creating a time index and results in a lot of details or is smoothed out, depending on the window length.

3.2 Feature Extraction

When extracting features, we strive to extract noise-robust features from samples produced under different noise environments since noise corrupts samples and distorts features. This work is part of the

"Front end" of ASR. As for the "back end, " it's a process of modifying the speech model parameters or changing them altogether in order to properly deal with any noise related distortions. Speech samples are processed into speech feature vectors after being sampled at the Nyquist rate. The use of 100 feature-vectors per second with tens of feature components per vector is typical; however, it does depend on the system. Extracted features need to be discriminative to discriminate classes from each other (sufficiently separated in a representation space) while being robust to variations within the same class. Common features are Linear Predictive Coefficients (LPCs), Mel-Frequency Cepstrum Coefficients (MFCCs), and ones that are based on Artificial Neural Networks (ANNs). MFCCs are not very robust with additive noise, thus it is common to normalize their values in speech recognition to lessen the noise influence [15], [16]. After extraction, features are directed to the decoder for proper recognition.

3.3 Hidden Markov Models (HMMs)

Voice signals have temporal structure, and the ASR system's job is to turn variable-length speech utterances into variable-length word sequences. Since speech is a sequence of sounds with properties that vary with respect to time, we take short segments of speech (frames of speech) from which we extract certain characteristics that will help us recognize a sound identity. The signal processing phase to extract a set of features is called parameterization (see Figure 3.1).

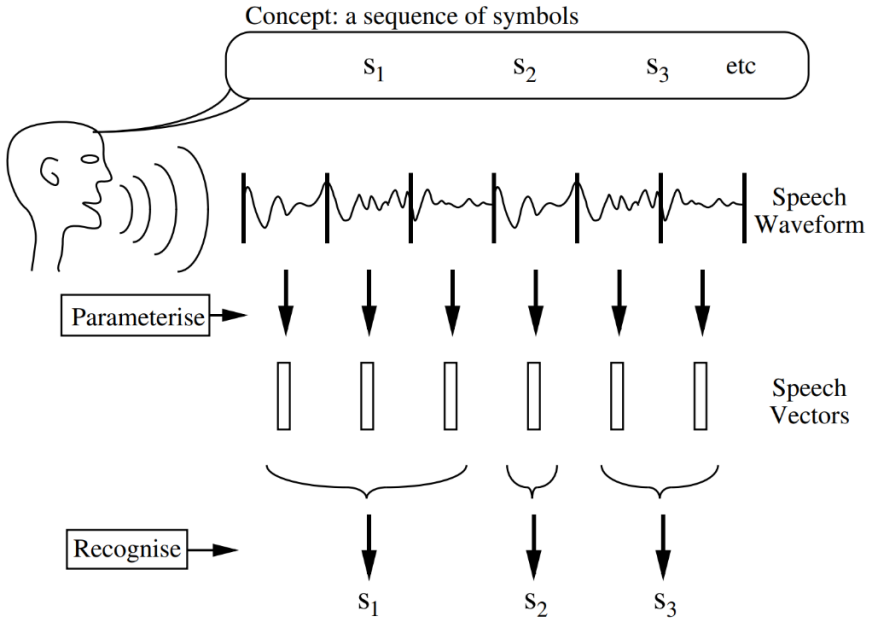


Figure 3.1: Message Encoding / Decoding [17]

Hidden Markov models (HMMs) provide a statistical framework for speech signal acoustic modelling. HMMs convert observation sequences (acoustic frames) into label sequences (phonemes). HMMs offer the probability distribution across all potential label sequences for a given acoustic observation [18]. Speech is highly non-stationary and varies with time; however, HMMs model speech assuming two circumstances: quasi-stationary and conditional independence [19], [20]. Despite the fact that feature vectors are highly correlated, conditional independence assumes feature vectors are conditionally independent of ones that are before and after with the next state transition probability depending only on the current state. Figure 3.4 illustrates a typical HMM model as part of a DNN-HMM structure. Parameters that need to be estimated for the HMM-based acoustic model are [20]:

- π: Initial state distribution. Initial state probability:

$$\boldsymbol{\pi_i} = \boldsymbol{P}(\boldsymbol{\phi_0} = \boldsymbol{s_i}) \tag{3.1}$$

  where ϕ_t: the state at time, t.

- $\prod = \{\pi_1, \pi_2, \dots \pi_N\}$ is a valid distribution if:

$$\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{\pi_i} = \mathbf{1} \tag{3.2}$$

  where N is the number of observation models in a mixture model.

- $\mathcal{A}$ is a State transition probability matrix from state s_i to state s_j, $\mathcal{A} = \{a_{ij}\}$

$$\boldsymbol{a_{ij}} = \boldsymbol{P}(\boldsymbol{\phi_t} = \boldsymbol{s_j} | \boldsymbol{\phi_{t-1}} = \boldsymbol{s_i}) \tag{3.3}$$

  When $\mathcal{A}$ is a full matrix, the transition probabilities satisfy:

$$\sum_{\boldsymbol{j=1}}^{\boldsymbol{N}} \boldsymbol{a_{ij}} = \mathbf{1} \tag{3.4}$$

  ß is the probability distribution of state output. ß $= \{b_1(\boldsymbol{o}_t), b_2(\boldsymbol{o}_t), b_3(\boldsymbol{o}_t) \ \dots, b_N(\boldsymbol{o}_t)\}$: The PDFs of acoustic feature vectors, given the states, where the PDF or likelihood of acoustic feature vector $\boldsymbol{o}_t$ in a state s_i at time t is:

$$\boldsymbol{b_i}(\boldsymbol{o_t}) = \boldsymbol{p}(\boldsymbol{o_t} | \boldsymbol{\phi_t} = \boldsymbol{s_i}) \tag{3.5}$$

The likelihood or probability distributions can be calculated by a Gaussian Mixture Model (GMM) or by a Deep Neural Network (DNN) given their popularity as acoustic models computing likelihoods in ASR systems. This results in hybridized models, GMM-HMM and DNN-HMM. The probability distributions of voice utterances linked with the states of HMMs were previously estimated using Gaussian mixture models (GMMs) [21]. The maximum likelihood technique was used to train acoustic models based on GMMs-HMMs. The sequence discriminative algorithm took over from the maximum likelihood approach in the 2000s. ASR's performance accuracy was further enhanced using sequence discriminative techniques such as minimal classification error and minimum phone error. GMMs have recently been replaced with discriminative hierarchical models such as deep neural networks (DNNs), which have increased the accuracy of the ASR system dramatically [22]. The availability of a big quantity of data and processing resources is a primary driving force for these discriminative hierarchical models.

3.4 Gaussian Mixture Models (GMMs)

GMMs are a probabilistic notion that are used to categorise data based on its probability distribution. GMMs may be used to represent any data set using multiple Gaussian distributions. In addition, they may be used to identify clusters in data sets when the clusters are not well defined. The Gaussian mixture model is a probabilistic model in which all data points are assumed to be created using a mixture of Gaussian distributions. Moreover, GMMs may be used to determine the likelihood that a new data point is a member of each cluster. Additionally, GMMs are relatively immune to outliers, which means that they may still provide accurate findings even if some data points do not neatly fit into any of the clusters [23]. As a result, GMMs are a versatile and strong tool for data clustering. It may be thought of as a probabilistic model in which each group is believed to have Gaussian distributions and has means and covariances that determine its parameters. GMMs are composed of two components: mean vectors and covariance matrices. A Gaussian distribution is a continuous probability distribution that assumes the form of a bell curve.

GMMs are often used to describe the probability distribution of continuous measurements or characteristics in a biometric system, such as vocal-tract related spectral parameters in a speaker identification system. GMM parameters are estimated from training data using the iterative Expectation-Maximization (EM) approach or from a well-trained prior model using Maximum A Posteriori (MAP) estimation [24]

3.5 Deep Neural Networks (DNNs)

DNNs are another alternative of state output distribution with the ability to model complex non-linear relationships. DNNs have the capacity to derive discriminative internal representations from speech signals that are resilient to the several sources of variability. With increasing network depth, these representations become less susceptible to small input perturbations, resulting in improved speech recognition performance [25]. DNNs are typically feedforward neural networks, as illustrated in Figure 3.2. They can however, be trained via backpropagation, enabling us to compute and assign errors associated with each neuron. This enables us to effectively alter and fit the model's parameters.

To assess the model's accuracy as it is being trained, a cost function is used. The cost function needs to be minimized so as to guarantee the model fits each observation correctly. While adjusting the model's weights and bias, it makes use of the cost function and reinforcement learning to achieve the point of convergence, or local minimum. The method modifies its weights using gradient descent, which enables the model to find the best direction to take in order to minimize mistakes, i.e., to minimize the cost function. The model's parameters are adjusted with each training case in order to progressively converge on the minimum. DNNs must take into account a variety of training factors, including number of layers, units per layer, learning rate, and the initial weights. With a high number of parameters, deep neural networks are very powerful machine learning systems [26]. However, overfitting is a significant issue with these networks. Dropout is a strategy for resolving this problem [27]. During training, the basic concept is to randomly remove units, dropping them with their connections from the neural network [28], [29]. This effectively eliminates overfitting.

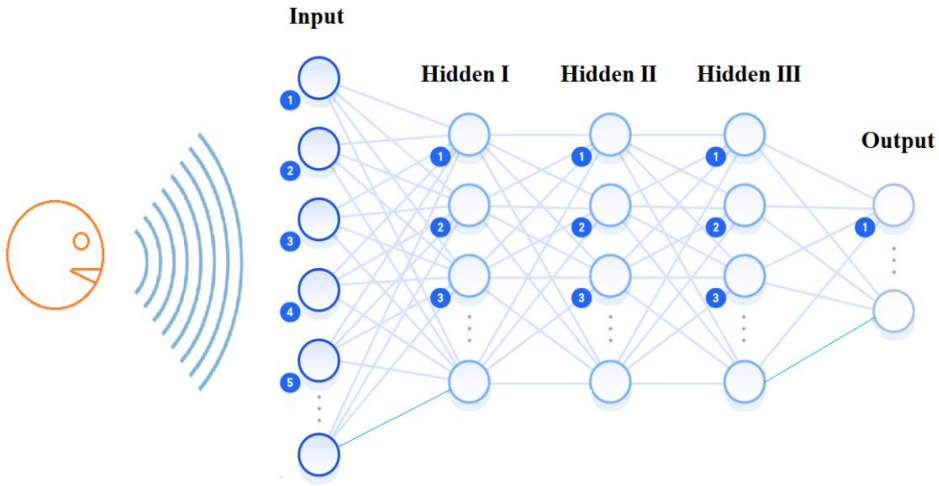


Figure 3.2: DNN Structure

The general mathematics behind a DNN is as follows:

1. Parameters (Weights $\boldsymbol{W}$and Bias $\boldsymbol{b}$ are randomly initialized. During the forward propagation process, the input neurons are multiplied by the weights and passed on to an activation function:

$$\boldsymbol{Z} = \boldsymbol{f}(\boldsymbol{W} \cdot \boldsymbol{X} + \boldsymbol{B}) \tag{3.6}$$

where Z denotes the output of a layer, X is the input, and f is a nonlinear activation function such as the sigmoid, hyperbolic tangent (tanh), or the rectifier linear unit function (ReLu) [30]:

Sigmoid activation function:

$$\boldsymbol{\sigma}(\boldsymbol{x}) = \frac{\mathbf{1}}{\mathbf{1} + \boldsymbol{e}^{-x}} \tag{3.7}$$

tanh activation function:

$$\boldsymbol{tanh}(\boldsymbol{x}) = \frac{\boldsymbol{e}^{x} - \boldsymbol{e}^{-x}}{\boldsymbol{e}^{x} + \boldsymbol{e}^{-x}} \tag{3.8}$$

ReLu activation function:

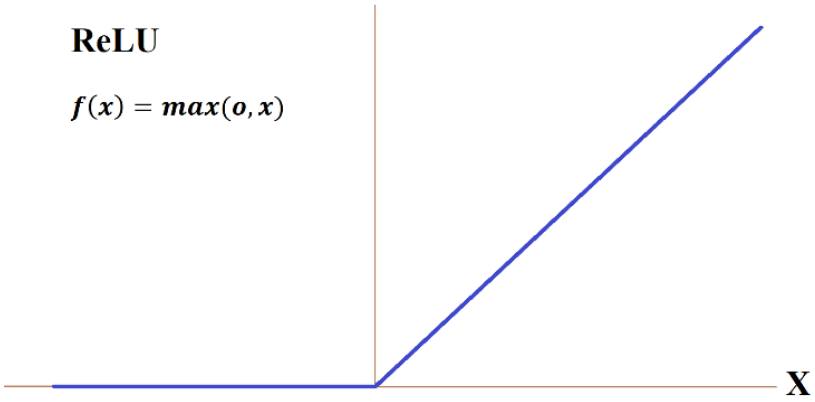


Figure 3.3: ReLU activation function

Activation functions provide non-linearity into the network, allowing it to learn complex patterns in input, such as photos, text, videos, or sounds. Without an activation function, Neural Networks would have minimal ability to learn. The activation function is a straightforward mathematical operation that converts the provided input to the desired output with a specified range. It activates the neuron when the output surpasses the function's specified threshold value. Activation functions are essentially responsible for turning the neuron ON and OFF. The neuron is supplied with the sum of its inputs and randomly initialized weights, as well as a static bias for each layer.

The activation function is applied over the sum forming an output. At the output layer, the softmax function (softargmax function) is applied, which reduces a vector of *K* output values to real values that sum up to one. These values will be between 0 and 1, so as to be interpreted as probabilities. The softmax function can be written as:

$$\boldsymbol{\sigma}\,(\vec{\boldsymbol{z}})_{\boldsymbol{i}} = \frac{\boldsymbol{e}^{\boldsymbol{z}_i}}{\sum_{\boldsymbol{j}=\boldsymbol{1}}^{\boldsymbol{K}} \boldsymbol{e}^{\boldsymbol{z}_j}} \tag{3.9}$$

where $\vec{\boldsymbol{z}}$ is the softmax function's input vector, and $\boldsymbol{z_i}$ are its elements, and $\boldsymbol{K}$ is the number of classes.

A loss function enables a neural network to identify the degree to which its predictions are incorrect. As the loss function returns the error associated with a single data point (sample), the cost function returns the error associated with the whole dataset. A neural network's cost is equal to the total of its losses on individual training samples. The loss and cost function can be written as:

$$\boldsymbol{Loss} = -\sum_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{k}} [\boldsymbol{y_i} \cdot \boldsymbol{log}(\boldsymbol{a_i})] \tag{3.10}$$

$$\boldsymbol{Cost} = -\frac{\boldsymbol{1}}{\boldsymbol{m}} \sum_{\boldsymbol{j}=\boldsymbol{1}}^{\boldsymbol{m}} \sum_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{k}} [\boldsymbol{y_i} \cdot \boldsymbol{log}(\boldsymbol{a_i})] \tag{3.11}$$

where $\boldsymbol{n}$ is the number of neurons in the output layer, $\boldsymbol{m}$ is the total datapoints, $\boldsymbol{a_k}$ is the value of the neuron ($a_k{}'th$ neuron). $\boldsymbol{y_k}$ is the value of the actual output label.

During backward propogation, the weights and bias are updated starting from the last layer and back as follows :

$$\boldsymbol{W_i} = \boldsymbol{W_i} - \boldsymbol{\alpha} \cdot \frac{\boldsymbol{\partial Cost}}{\boldsymbol{\partial W_i}} \tag{3.12}$$

$$\boldsymbol{B_i} = \boldsymbol{B_i} - \boldsymbol{\alpha} \cdot \frac{\boldsymbol{\partial Cost}}{\boldsymbol{\partial B_i}} \tag{3.13}$$

where $\boldsymbol{w_i}$ denotes the weights of layer $\boldsymbol{i}$, $\boldsymbol{\alpha}$ denotes the learning rate, and $\boldsymbol{B}$ represents the bias.

3.6 Hybrid DNN-HMM

Without a doubt, Deep Neural Network Hidden Markov Models (DNN-HMMs) have shown they can outperform Gaussian mixture model-based Hidden Markov Models in terms of speech recognition (GMM-HMMs) [31]. The DNN-HMM hybrid system as illustrated in Figure 3.4 leverages the representation learning capability of DNN and the sequential modelling capability of HMM on a variety of big vocabulary continuous audio recognition tasks [24].

The hybridization of DNNs & HMMs has had promising results in both speech and image recognition unlike in early stages where neural networks substituting GMMs consisted of just a single hidden layer making it difficult to adapt, and taking too long to train. Observations from multiple consecutive frames are fed as input features into a sequence of hidden layers [20].

HMMs are used to represent the speech signal's dynamics, while DNNs are used to estimate the observation probabilities. Each DNN output neuron is trained to predict the posterior probability of an HMM's state continuous density given the acoustic data. The DNN model converts speech features to labels corresponding to hidden Markov states in Hidden Markov Models (HMMs) [32]. Apart from its intrinsic discrimination, DNN-HMMs offer two additional advantages: training may be carried out using the incorporated Viterbi algorithm, and decoding is typically highly efficient [24].

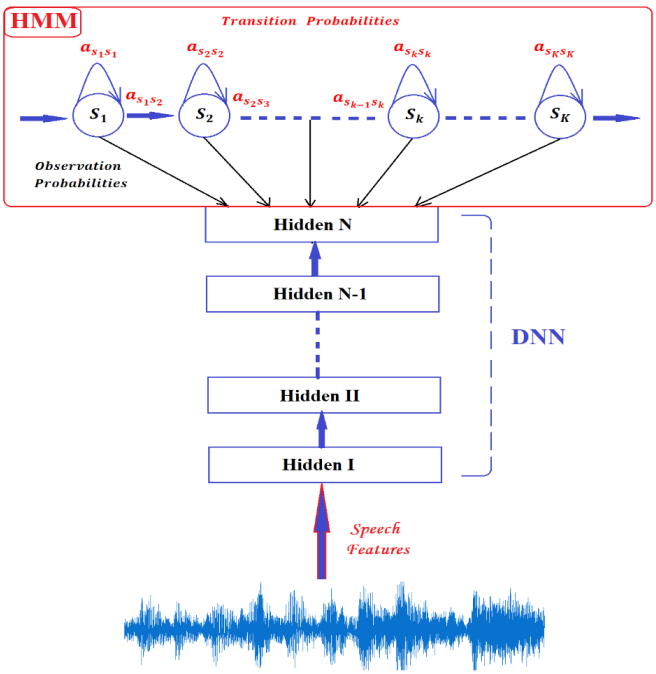


Figure 3.4: Hybrid DNN-HMM Structure

3.7 Convolutional Neural Networks (CNNs)

CNNs are widely used in artificial intelligence as a result of the significant advances they have made, most notably in the processing of sequential data sets. The popularity of CNNs arises from their ability to extract a hierarchy of robust features automatically, hence boosting performance. The structure of a conventional neural CNN is comprised of one or more convolutional layers, a max pooling layer, and a fully connected layer followed by backpropagation as illustrated Figure 3.5: CNN Architecture. The structure is composed of many neurons, with multiple identical clones of the same neuron in each convolutional layer, which provides CNNs with the benefit of not requiring a large number of actual parameters for big model calculations.

The primary benefit of CNNs is that they significantly reduce the number of parameters in ANNs [1]. The disadvantage of ANN is the large number of weight parameters that must be trained. When training on a colourful picture with RGB channels of size *500 by 500 px*, the total number of input parameters is *500 x 500 x 3 = 750,000*. If the first hidden layer has 100 neurons, the total number of weight parameters is *750,000 x 100 = 75,000,000*, which is much too numerous to train. As a result, training becomes a time-consuming process.

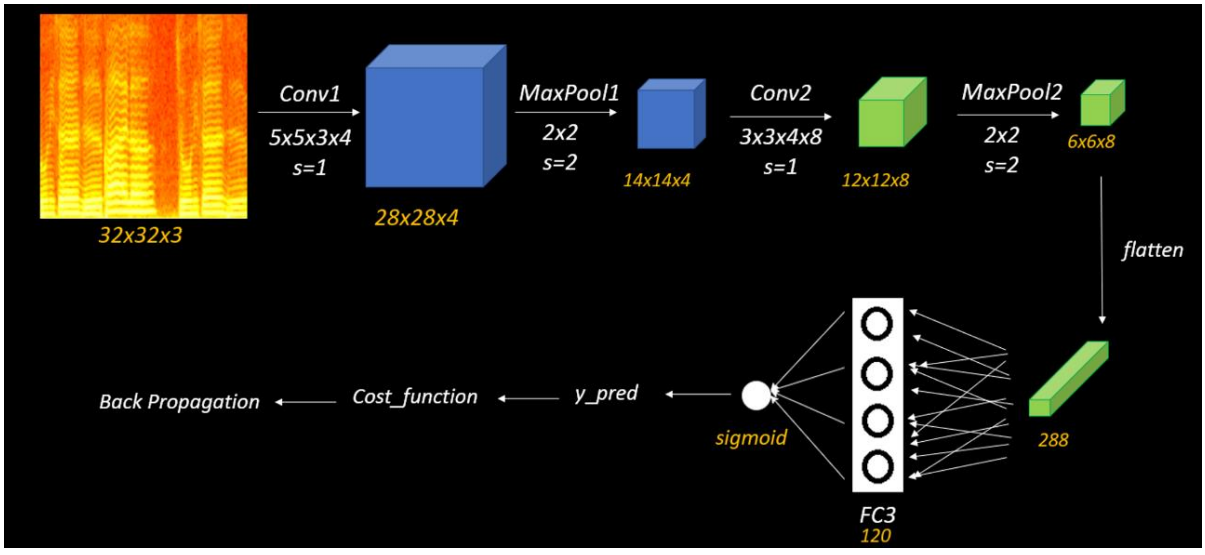


Figure 3.5: CNN Architecture

The core concept of convolutional neural networks is the use of filters. A filter is a matrix that acts as a sliding window inside an image and is responsible for recognising features or patterns within images using convolution. Rather than examining the entire image, filters look for specific sections [33]. An image has

edges, forms, and colours, and when all of these are combined, it has features [34]. Filters convolve the input and transmit the resulting output to a pooling layer. Again, the pooling layer's output may be routed via a mix of convolutional and pooling layers. Finally, it is transmitted to a layer that is fully connected.

3.7.1 Convolutional layers

During the convolution process, images are convolved with filters which are also referred to as kernels. The kernel may be thought of as superimposed on an image, with the components in the overlapping pixels multiplied and added together to provide a value. The kernel is then shifted forward by a certain amount, referred to as the "stride." The convolutional output is sent to an activation function, commonly, the ReLU activation function. Convolution reduces the dimension size of a ***n×n*** picture to:

$$(n \cdot n) * (f \cdot f) = \left(\frac{n-f}{s} + 1\right) \cdot \left(\frac{n-f}{s} + 1\right) \tag{3.14}$$

As illustrated in Figure 3.6, if K_{ij} represents kernel values and I_{ij} represents image pixels values, then the output of the convolved image can be given by equation (3.15). The computed output of the example in Figure 3.6 is represented in equation (3.16) as follows:

$$C_{ij} = \sum K_{ij} * I_{Ij} \tag{3.15}$$

$$c_{11} = 1 \cdot 1 + 2 \cdot 1 + 3 \cdot 1 + 6 \cdot 0 + 5 \cdot 0 + 7 \cdot 0 + 9 \cdot (-1) + 1 \cdot (-1) + 4 \cdot (-1) = -8 \tag{3.16}$$

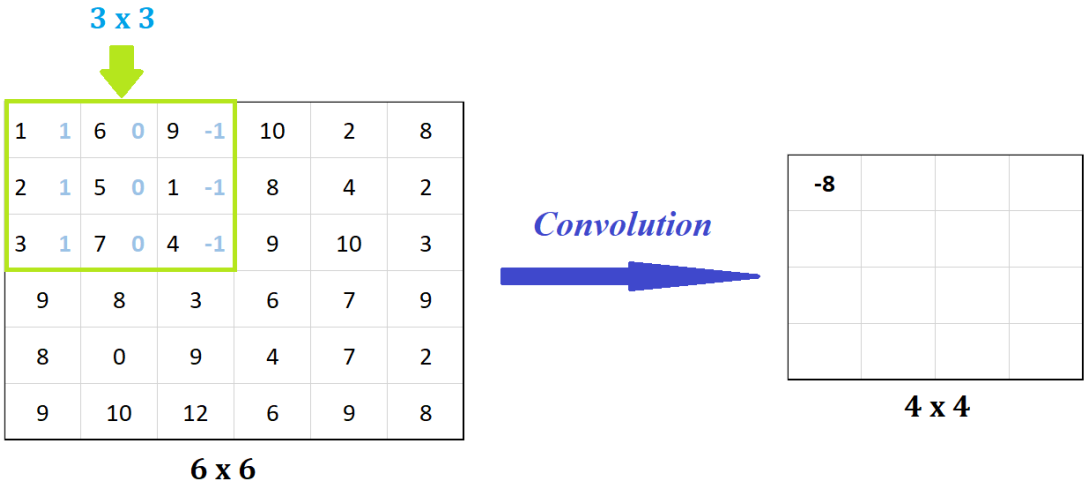


Figure 3.6: Convolution Process

3.7.2 Pooling Layer

Pooling's primary objective is to reduce the complexity of subsequent layers by down-sampling. In terms of image processing, this is analogous to decreasing the resolution. Pooling has no effect on the number of filters. Max-pooling is a common kind of pooling strategy. It divides the image into rectangular sub-regions and delivers the greatest value of each sub-region. Figure 3.7 illustrates the impact of max pooling.

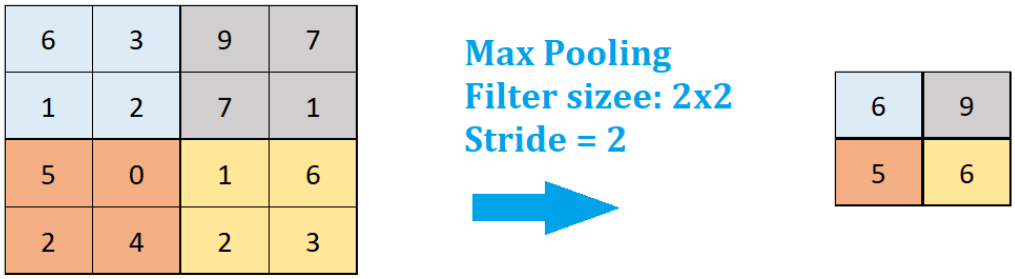


Figure 3.7: Max Pooling

3.7.3 Spectrograms and CNNs

Convolutional Neural Networks can discriminate spectro-temporal patterns, capturing patterns across time and frequency for given input spectrograms. They have an edge over MFCCs as they are still able to make distinctions even when sound is masked in time or frequency by other sound. It has been shown that spectrograms can be processed as images and that CNNs can be used to accomplish neural style transfer. [35],[36]. Spectrograms are two-dimensional representations that depict sequences of spectra along one axis in time, and frequency along the other, with brightness or colour denoting the intensity of a frequency component at each time frame. Thus, this representation suggests that some of the convolutional neural network designs developed for images might be simply adapted to sound [37].

3.7.3.1 Challenges

1. Due to the fact that image processing networks operate on three-channel RGB input, the magnitude values of the spectrograms on a single channel must be replicated over three channels in order to interact with the pre-trained network.

2. CNNs learn location invariant features. However, horizontal displacement of a sound event alters its temporal location, and one may argue that a sound event implies the same thing regardless of when it occurs. Vertically shifting a sound, on the other hand, may alter its meaning. Frequency shifting a sound event also alters its spatial breadth.
3. While comparable surrounding pixels in images may often be presumed to correspond to the same visual object, in sound, frequencies are frequently dispersed non-locally on the spectrogram [37]. Periodic sounds are often composed of a fundamental frequency and a number of harmonic frequencies that are separated by relationships determined by the sound's source. The timbre of a sound is determined by the combination of these harmonics.

   For example, In the case of a female voice, the fundamental frequency may be 200Hz at any one time, whereas the first harmonic is 400Hz, the second harmonic is 600Hz, and so on. These frequencies are not spatially clustered, but rather move in lockstep with one another due to a shared connection. This complicates the challenge of detecting local features in spectrograms using 2D convolutions, since they are often irregularly spaced apart, despite the fact that they flow in the same manner [38].

3.8 One Dimensional Convolutional Neural Networks (1D CNNs)

As discussed previously, CNNs are optimised for two-dimensional data. A 1D CNN is a more appropriate model for 1D data, such as audio signals. In 1D CNNs, the kernel moves in a single direction, resulting in one-dimensional input and output data, whereas in 2D CNNs, the kernel moves in two directions [39]. For 1D series data such as audio signals, a 1D CNN outperforms a normal 2D CNN. A 1D CNN's overall design as illustrated in Figure 3.8 is very much like that of a 2D CNN. The one-dimensional input data is passed to a mixture of one-dimensional convolutional and pooling layers. The last pooling/convolutional layer's output is then sent to fully connected layers, which provide a classification-ready final output.

Not only does a 1D CNN need fewer parameters to train, but the computational cost is much lower than that of a 2D CNN [39]. When an image of size *NxN* is convolved with a kernel of size *KxK* using a two-dimensional convolutional neural network, the computational complexity approaches $O(N^2K^2)$. On the other hand, the computational cost of dealing with *N*-dimensional series data and convolving with a *K*-dimensional kernel is around *O(NK)*, which is much less. Additionally, for each kernel size, the number of parameters to train is decreased to *K* from K^2. As a result, a 1D CNN is well-suited to dealing with time series represented by audio data signals.

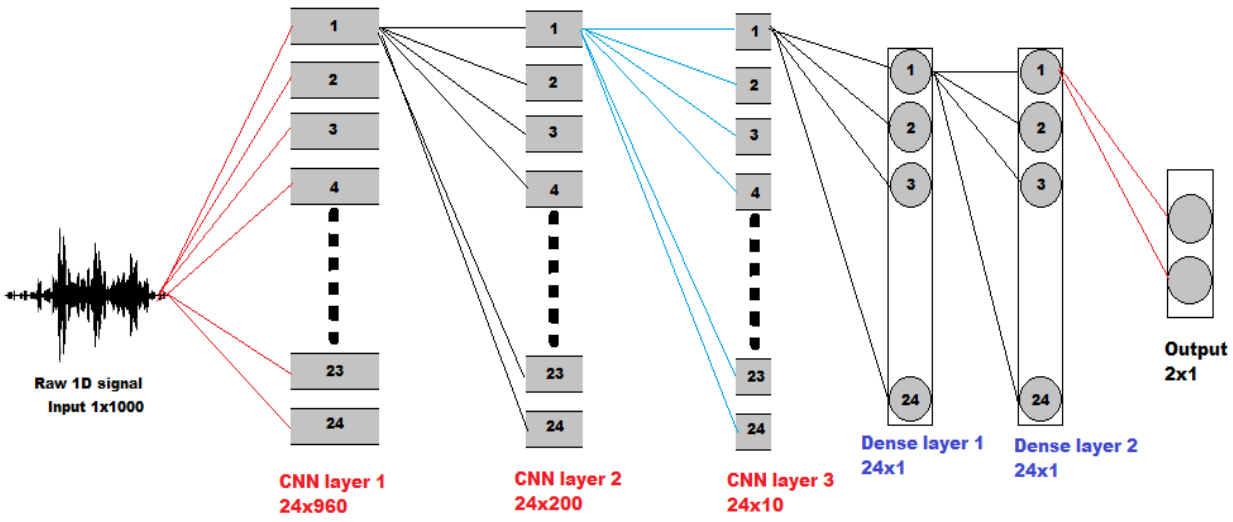


Figure 3.8: 1D-CNN design example

3.9 WaveNet

WaveNet is a fairly new concept with great potential. It has the ability to produce genuine sounding speech imitating a human's voice. Per [40], WavNet is a convolutional deep neural network that is used to generate raw audio waveforms. WaveNet is capable of generating speech that sounds more authentic and accurately mimics any human voice than the finest existing Text-to-Speech systems. Prior to WaveNets, statistical speech synthesis or text-to-speech mostly relied on concatenative TTS, in which a large database of speech fragments from a speaker is captured and recombined to generate whole utterances.

This makes altering the characteristics of a voice very difficult without generating a new database; as in changing the emotion or emphasis of speech. WaveNet however, disrupts this paradigm by modelling the raw waveform of an audio stream one discrete sample at a time. WaveNet can replicate any sort of audio, resulting in more natural-sounding speech. A single WaveNet can accurately record the characteristics of a wide range of speakers and transition between them depending on the speaker's identification.

To cope with long-range temporal dependencies required for raw audio creation, WaveNet is structured around dilated causal convolutions with very broad receptive fields. Dilated convolution may be thought of as convolution with filters applied across an area larger than its length by omitting some input values. It enables the network to function on a rudimentary scale. This may be compared to pooling or striding. The rationale is that the outcome of a single time step is able to depend on a longer input sequence of

preceding time step inputs [41]. The model operates directly on raw audio waveforms. Each audio sample x_t is conditioned on all previous timesteps by factorizing as a product of conditional probabilities as represented in equation (3.17). The output of the model is a categorical distribution over the next value of x_t.

$$p(x) = \prod_1^T p(x_t|x_1, \dots, x_{t-1}) \tag{3.17}$$

3.10 Long Short-Term Memory (LSTM)

A Long Short-Term Memory (LSTM) Network is a recurrent neural network (RNN), which is basically a network with loops. RNNs have shown great success in speech recognition, language modelling, translation, and more. Unfortunately, RNNs don't learn well in connecting information when there exist very long time lags [42], as they suffer from vanishing gradient descent during backpropagation, thus dubbed as having short memory. This is where LSTM networks come in. They are a special kind of RNN that is capable of learning long-term dependencies. LSTMs, as the name suggests, is able to retain both short and long memory.

The ability to remember information for long periods of time is a default behavior of LSTM networks. A LSTM network is able to classify, process, and predict time series when there are very long time lags of unknown size between events, outperforming RNNs and HMMs. [43] With respect to automatic speech recognition (ASR), LSTMs achieved a record 17.7 % phoneme error rate on the classic TIMIT natural speech dataset back in 2013 [13].

A repeating model in LSTM contains four interacting layers. Figure 3.9 illustrates an LSTM cell design. The three gates shown in the diagram are there to protect and control the cell state. Through them; Information is added or removed to the cell state. The gates are a sigmoid neural net layer that carry a value of "1" or "0" where the value "1" lets information through and "0" does the opposite.

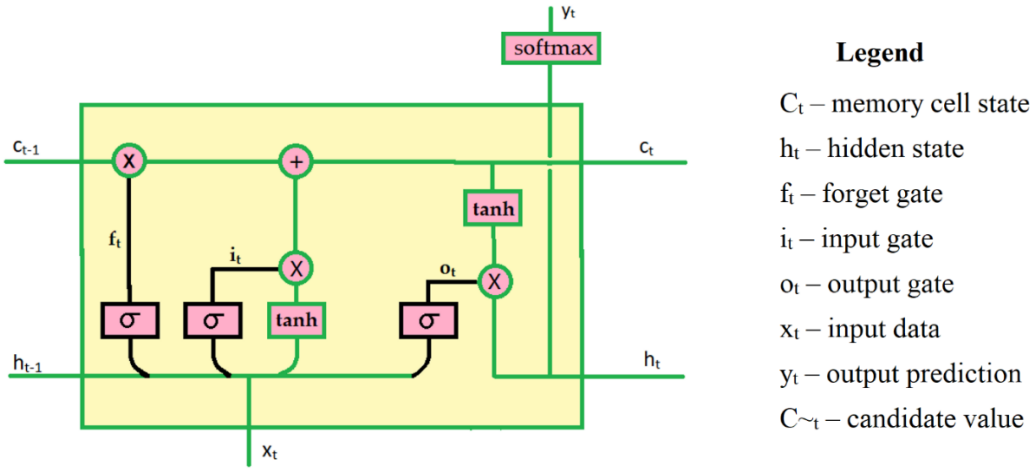


Figure 3.9: LSTM Cell design

3.10.1 How LSTM Works

The following steps show how the gates operate:

1. The first sigmoid layer is the “forget gate”. It allows information to be retained or removed from the cell state.
2. The second sigmoid layer is the “input gate”. It decides what information is to be stored or added to the cell state. In 2014, the idea of combining the “forget” and “input” gate into a single “update gate” was introduced.
3. The “tanh” layer creates a vector of new values to add to the cell state.
4. Here, steps 2 and 3 are combined to create an update to the state.
5. The last sigmoid layer decides what parts of the cell state will be the output.
6. Finally, the cell state is put through “tanh” and multiplied by the output of the last sigmoid layer.

Figure 3.10 illustrates the unit architecture of LSTM. We notice the horizontal “cell state” line, which is the key to LSTM. This cell state line runs across the entire chain with minor linear interactions. LSTMs can perform sophisticated, artificial long-time-lag challenges that prior recurrent network algorithms have never been able to accomplish [44].

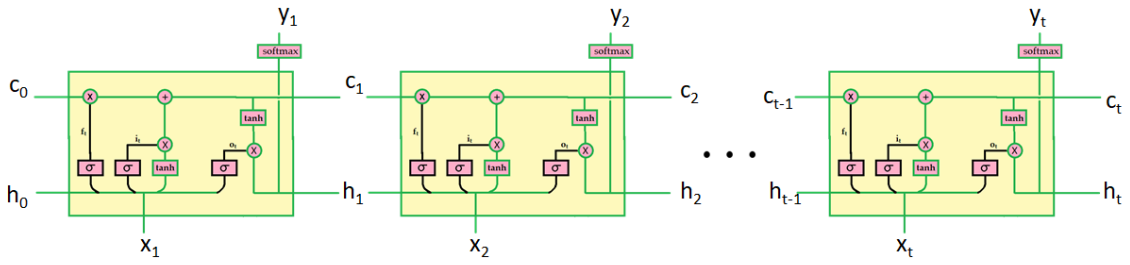


Figure 3.10: LSTM unit-architecture

3.11 Hyper-Parameters in Machine Learning Algorithms

Machine learning models have parameters that are trainable and customizable. Trainable parameters are those whose values are altered automatically during the model's training process, while tunable parameters are tuned or adjusted manually or via the use of an automated tuning approach prior to the model's training. Trainable parameters include the weights and bias of a neural network. On the other hand, configurable or tunable parameters include learning rates, epochs, and so on. In machine learning, these customizable parameters are referred to as hyperparameters.

In this respect, hyperparameters are considered to be external to the model since their values cannot change during learning/training. They are established prior to the start of the training phase. While they are utilized to train the machine learning model, they are not included in the resultant model. The trainable parameters, on the other hand, might be referred to as a resultant model.

Hyperparameter configuration of an algorithm, in many cases, affect its performance on a specific learning job. Experts in machine learning can fine-tune the hyperparameters to optimise the results [45]. Lavesson and Davidsson [46] assert that hyperparameter tweaking is often more crucial than the machine learning technique employed. Hyperparameters have a significant impact on the performance of any machine learning model on a given dataset [47]. Appropriate hyperparameter optimization may result in a large performance boost.

The basic issue with hyperparameters is that it is not always evident which ones are critical to modify and which ones will have little effect on model performance [45]. Hyperparameter tweaking is a computationally demanding operation that relies on the dataset utilized. The combination of hyperparameter values that perform well on one dataset is not necessarily likely to work well on another. As a result, adjusting the hyperparameters is a job on its own.

3.12 Kaldi Toolkit

Kaldi is a free and open-source toolbox for speech recognition researchers. The toolkit was first created in 2009 at John Hopkins University by Daniel Povey and colleagues. It is developed in C++ and is extensible. Kaldi provides significant support for linear algebra, and can generate features such as MFCCs, fbank, and fMLLR. Thus, in current research on deep neural networks, Kaldi is often used to pre-process raw waveforms into acoustic features for end-to-end neural models. For a more detailed examination of Kaldi, please refer to Appendix B.

This Page Intentionally Left Blank

Chapter 4

Related Work to Enhance ASR by Data Augmentation

The main themes for data-augmentation in recent works is to avoid direct manipulation of the speech data, but rather to do some well formulated mapping from one speech domain to the another. The neural networks described in the previous chapter became recently the most powerful way to find this type of complex mapping. The following is a summary of the main related works.

4.1 Convolutional-Augmented Transformer for Speech Recognition

4.1.1 Idea and Comparison

Combining CNNs with Transformer architecture based on self-attention to model local and global dependencies result in a distinctive Augmentive Conformer block (Convolution-augmented transformer) [48]. The Conformer block encompasses a stack of a feed-forward model, a self-attention model, a convolution model, and ends with a second feed-forward model. The convolution model being stacked after the self-attention model is the most suitable for speech recognition. Other research methods adopt squeeze-and-excitation model to capture longer context, but is still limited in capturing global context. This combining idea was inspired by “Wu et al” [49], [50], who initially proposed splitting the input into two branches: self-attention and convolution; and then concatenating their outputs.

4.1.2 Improvements and Drawbacks

Conformer achieves an impressive state-of-the-art-results on “LibriSpeech” with a 15% improvement over the best published Transformer Transducer on the test-other dataset with an external language model. The Conformer achieves the lowest WER with the addition of a language model. The conformer showed interesting results against other models. Table 4.1 compares the outcome of the conformer with other models. There isn’t a real downside, but some specifications have to be applied such as: The stack has to be in a specific order with two FFNs sandwiching the self-attention model and the convolution model. Increasing the number of attention-heads up to 16 gives best accuracy, and the convolutional kernel sizes give best performance up to 17 and 32 [48].

Table 4.1: Conformer model on LibriSpeech [48]

Method	Params (M)	WER without LM		WER with LM	
		Test clean	Test other	Test clean	Test other
Hybrid					
Transformer [51]	-	-	-	2.26	4.85
LAS					
Transformer [52]	270	2.89	6.98	2.33	5.17
Transformer [53]	-	2.2	5.6	2.6	5.7
LSTM	360	2.6	6	2.2	5.2
Transducer					
Transformer [54]	139	2.4	5.6	2	4.6
ContextNet (S) [55]	10.8	2.9	7	2.3	5.5
ContextNet (M) [55]	31.4	2.4	5.4	**2**	4.5
ContextNet (L) [55]	112.7	**2.1**	4.6	**1.9**	4.1
Conformer					
Conformer (S)	10.3	**2.7**	**6.3**	**2.1**	**5**
Conformer (M)	30.7	**2.3**	**5**	**2**	**4.3**
Conformer (L)	118.8	**2.1**	**4.3**	**1.9**	**3.9**

4.2 SpecAugment: Data Augmentation method for ASR

4.2.1 Idea and Comparison

Per [56], this data augmentation method is applied directly to a neural network's feature inputs like the filter bank coefficients. In this method, SpecAugment is applied on Listen, Attend, and Spell networks (LAS) end to end [57], outperforming all previous work on LibriSpeech 960h and Switchboard 300h. Table 4.2 and Table 4.3 display the results on Librispeech 960h and Switchboard 300h respectively compared with the closest results of other models. SpecAugment operates on the log Mel spectrogram of the input audio and not the raw audio, as if it's an image. It consists of three deformations; Time warping, Time masking, and Frequency masking. Despite the simplicity of this method, it is very effective and low in computational cost. It has shown state of the art performance even without a Language Model (LM). Learning rate schedule along with augmentation can maximize network performance. In this augmentation method, both long and very long schedules are introduced, and a RNN language model by shallow fusion [58][59] is adopted.

4.2.2 Improvements and Drawbacks

The Results out-perform previous hybrid systems. State of the art performance is achieved even without a language model (see Table 4.2 and Table 4.3). In addition, there is a major benefit to training with augmentation; networks under-fit the loss and WER, converting an over-fitting problem into an under-fitting problem. This is one way of regularizing and model with excessive capacity than needed and make it learn the proper generalization of the problem rather than memorizing it. When addressing under-fitting by standard approaches, significant gains in performance were achieved.

Table 4.2: SpecAugment on LibriSpeech 960h WERs (%) [56]

LibriSpeech 960h WERs (%)				
Method	**No LM**		**With LM**	
	clean	other	clean	other
HMM				
Yang et al., 2018	-	-	2.97	7.5
CTC/ASG				
Li et al., 2019 [60]	3.6	11.95	2.95	8.79
LAS				
Irie et al., 2019 [61]	4.7	13.4	3.6	10.3
Sabour et al., 2019	4.5	13.3	-	-
SpecAugment				
LAS	4.1	12.5	3.2	9.8
LAS + SpecAugment	**2.8**	**6.8**	**2.5**	**5.8**

Table 4.3: SpecAugment on Switchboard 300h WERs (%) [56]

Switchboard 300h WERs (%)				
Method	**No LM**		**With LM**	
	clean	other	clean	other
HMM				
Zeyer et al., 2018	-	-	8.3	17.3
LAS				
Weng et al., 2018 [62]	12.2	23.3	-	-
Zeyer et al., 2018 [63]	11.9	23.7	11	23.1
SpecAugment				
LAS	11.2	21.6	10.9	19.4
LAS + SpecAugment (SM)	**7.2**	14.6	**6.8**	14.1
LAS + SpecAugment (SS)	7.3	**14.4**	7.1	**14**

As a downside, it was concluded that time warping has no major impact on performance improvement, and since it is the most expensive augmentation in this method, it is recommended that it is dropped when facing budget limitations. Label smoothing is introduced only at the beginning for LibriSpeech 960h because it can destabilize training for smaller learning rates, whereas it is kept for Switchboard 300h throughout the training process.

4.3 Noisy Student Training

4.3.1 Idea and Comparison

The idea is to adopt for ASR an existing semi-supervised learning method that has shown image classification improvements known as “Noisy Student Training”. The original principle behind this method is iterative self-training that leverages augmentation. As this topic is very relatable to our study, it was important to study the approach of [64] among the other proposed methods in this chapter. As [64] employs and implements the below tasks, we get an insight view to understand the pros and cons of this method:

- Adaptive SpecAugment is introduced as the augmentation method that acts on the spectrogram of audio input, employing adaptive time masking.
- Language Model Fusion on teacher network, generating better transcripts for student network to train. A normalized filtering score is proposed for the transcripts.
- Sub-modular sampling to balance the dataset by weighing the samples generated by the teacher network (utterance-transcript pairs).
- A normalised filtering score for transcripts created by teacher networks that is proportional to the fusion score and token count.

In addition, key methods were used to filter, balance, and augment the data that is generated in between self-training iterations, giving an improved WER. Noisy Student Training makes use of unlabeled data for improving accuracy. A series of models are trained in succession; a teacher – student relationship. The teacher is the preceding model in the series on the unlabeled data portion. The key feature is augmentation exploitation which enriches the training data with more variabilities enabling the ASR to generalize. The teacher produces quality labels from clean input, and the student reproduces them with heavily distorted augmented input features. The teacher produces quality labels from clean input, and the student reproduces them with heavily augmented input features.

4.3.2 Improvements and Drawbacks

To adapt the noisy student training pipeline for speech recognition, SpecAugment, language model fusion, and sub-modular sampling are employed. The first set of experiments were conducted on LibriSpeech 100-860, where the clean 100h subset is used as labelled data and the remaining 860h as the unlabeled data [65]. The results are represented in Table 4.4. The second set of experiments were conducted on LibriSpeech-LibriLight, in which the full Librispeech is used as labelled data and the unlab-60k subset as unlabelled data [66], Table 4.5 displays the results. It was concluded that the use of successive filtration for low-supervised performance LibriSpeech 100-860 was beneficial, whereas it was not much of a benefit to high-supervised performance LibriSpeech-LibriLight since the baseline model was already very good, and the quality of filtered transcripts does not differ much from the quality of the unfiltered transcripts.

Table 4.4: Noisy Student Training on LibriSpeech 100h WERs (%) [64]

LibriSpeech 100h WERs (%)				
Method	**Dev**		**Test**	
	clean	other	clean	other
Supervised				
Luscher et al., 2019 [67]	5	19.5	5.8	18.6
Semi-supervised (w/ LibirSpeech 860h)				
Hsu et al., 2019 [68]	5.39	14.89	5.78	16.27
Ling et al., 2019 [69]	-	-	4.74	12.2
Proposed Method				
Baseline (LAS + SpecAugment)	5.3	16.5	5.5	16.9
+ NST before LM Fusion	4.3	9.7	4.5	9.5
+ NST with LM Fusion	**3.9**	**8.8**	**4.2**	**8.6**

Table 4.5: Noisy Student Training on LibriSpeech 960h WERs (%) [64]

LibriSpeech 960h WERs (%)				
Method	**Dev**		**Test**	
	clean	other	clean	other
Supervised				
Zhang et al., 2020 [54]	-	-	2	4.6
Han et al., 2020 [55]	1.9	3.9	1.9	4.1
Semi-supervised				
Synnaeve et al., 2018 [52]	2	3.65	2.09	4.11
Proposed Method with baseline				
ContextNet + NST before LM Fusion	**1.6**	3.7	**1.7**	3.7
ContextNet + NST after LM Fusion	**1.6**	**3.4**	**1.7**	**3.4**

4.4 Text-To-Speech Data Augmentation

4.4.1 Idea and Comparison

When the available training data is large, hybrid DNN-HMM, and end-to-end neural network recognition perform nearly the same. However, when training data is not sufficiently large, end-to-end neural network recognition cannot come close to DNN-HMM performance. For that reason, data augmentation is significant in low-resource tasks. End-to-end neural network recognition needs data augmentation techniques such as morphing the training data set or including additional data, so as to be competitive. The proposed method in [70] uses speech synthesis for augmenting training data, which is highly relevant to our study since its major focus is on data augmentation.

The method of augmentation is a semi-supervised learning technique. For a 100-hour subset of training data, the TTS and baseline ASR systems are trained separately. Then, using a broader subset, utterances are synthesized and used as ASR training data. The Transformer from the ESPNET speech recognition toolkit is used for the ASR model. The Transformer is a sequence-to-sequence architecture consisting of two neural networks (Encoder and Decoder). The language model (LM) used is a recurrent neural network (RNN) consisting of four LSTM layers.

Utterances that are too long or too short are removed, and data is speed-perturbed, making the training set 3 times larger. Tacotron is selected as a base speech synthesizer, which creates a log-magnitude 80-band Mel-Spectrogram from the input test [71], [72]. Acoustic features are augmented with SpecAugment during the training phase. The TTS system is a two-network setup; a synthesizer (input text to spectrogram) and a vocoder (spectrogram to waveform).

4.4.2 Improvements and Drawbacks

When synthesized speech is added, the end-to-end ASR baseline achieved an improvement of 39% relative WER on test-clean and 21% on test-other. Table 4.6 shows the proposed method outperforms only training set clean-460 and outperforms others in the low-to-medium resource setup as shown in Table 4.7. The downside however is:

- Low-resource (training data amount) is a limitation.
- Although the system was superior in the medium-resource setup in WER, the improvement became less noticeable for test-clean and completely vanished for test-other which is the more challenging set from the LibriSpeech dataset.

Table 4.6: TTS Augmentation on LibriSpeech train-clean-100/460 [70]

Training Set	***ASR System***	***WER (%)***			
		dev		**test**	
		clean	other	clean	other
clean-100	Kaldi	5.9	20.4	6.6	22.5
	RETURNN Hybrids	**5**	**19.5**	**5.8**	**18.6**
	E2E proposed method	10.3	24	11.2	24.9
clean-460	Kaldi	5.3	17.7	5.8	19.1
	E2E proposed method	**4.5**	**14.1**	**5.1**	**14.1**

Table 4.7: TTS Augmentation in comparison with other works [70]

Setup	***Other works***	***WER (%)***		***WER Improvement (%)***	
		dev		**test**	
		clean	other	clean	other
Low-to-Medium Resource	**Proposed method**	**4.3**	**13.5**	**38.6**	**20.6**
	Method [73]	9.3	30.6	22.8	10.1
	Method [74]	5.4	22.2	33.3	9.4
Medium Resource	**Proposed method**	**3.2**	**9.1**	**8.6**	0
	Method [73]	6.3	22.5	0.3	-0.5
Large Resource	Method [75]	4.7	15.5	**8.6**	**4.6**
	Method [73]	4.6	13.6	4.6	1.8
	Method [74]	**2.5**	**7.2**	4.9	2.4

4.5 On The Fly Data Augmentation

4.5.1 Idea and Comparison

Sequence-to-Sequence (S2S) models are capable of state-of-the-art performance if overfitting is avoided. To do so, training data must be increased by data augmentation when data resources are low. [76] explores three augmentation methods that relate to our work in this thesis:

1. Dynamic Time Stretching: Input sequence is modified by manipulating the time series of the frequency vectors (features) to achieve the speed perturbation effect. Every feature vector window is stretched using nearest-neighbor interpolation.
2. SpecAugment: The spectrogram input is modified with frequency and time masking before being fed to the S2S model.
3. Sub-sequence Sampling (Constraint Sampling): Given an utterance, three different variants of sub-sequences are allowed; with equal distribution.

4.5.2 Improvements and Drawbacks

Two different sequence-to-sequence models are used: LSTM-based S2S and Self-Attention S2S. Both models can be improved by combining two augmentation strategies in a single training (for example, using Time Stretching first and then SpecAugment for input sequences). This suggests that both strategies aid in the generalization of models across various aspects and can be used in conjunction with one another. As represented in Table 4.8 and

Table 4.9, the combination of LSTM-based and self-attention models was found to be quite efficient for reducing WER. When no additional language models are used, state-of-the-art performance is attained on the Switchboard and CallHome (CH) test sets. With this proposed method, there are some drawbacks; LSTM-based S2S models are likely to overfit after 12k updates despite Dropout regularization. Overfitting occurs due to models being very large and deep, and Self-Attention models converge slowly, saturating at 40 k updates.

Table 4.8: On The Fly Augmentation on 300h Switchboard [76]

300h Switchboard			
Model	**LM**	**SWB**	**CH**
Zeyer et al., 2018	LSTM	8.3	17.3
Park et al., 2019	LSTM	7.1	14
Kurata et al., 2019	-	11.7	20.2
LSTM-based	-	8.8	17.2
Transformer	-	9	17.5
ensemble	-	7.5	15.3

Table 4.9: On The Fly Augmentation on 2000h SWB + Fisher [76]

2000h Switchboard + Fisher			
Model	**LM**	**SWB**	**CH**
Povey et al., 2016	n-gram	8.5	15.3
Saon et al., 2017	LSTM	5.5	10.3
Han et al., 2018	LSTM	5	9.1
Weng et al., 2018	-	8.3	15.5
Audhkhasi et al., 2018	-	8.8	13.9
LSTM-based (no Augmentation)	-	7.2	13.9
Transformer (no Augmentation)	-	7.3	13.5
LSTM-based	-	5.5	11.4
Transformer	-	6.2	11.9
ensemble	-	5.2	10.2

4.6 Low Resource Speaker Augmentation

4.6.1 Idea and Comparison

Because there are fewer speakers in low-resource tasks, this results in fewer speaker variations in synthetic speech. The idea here is to use a speaker augmentation technique to synthesize data with enough speaker and text diversity. [77] proposes a speaker augmentation scheme that trains an end-to-end speech synthesis model Tacotron2 [78], conditioned however on speaker representations from a variational autoencoder (VAE) [79]. In addition, a speaker classifier that uses latent variables as input is jointly trained. As a result, the audio encoder is more likely to output latent variables containing speaker information, which aids model convergence. The TTS model can synthesize speech from unknown new speakers using these methods, sampling from the taught latent distribution and delivering enough speaker variations in synthetic speech for data augmentation. The more virtual speakers are sampled, the better ASR performance.

4.6.2 Improvements and Drawbacks

When compared to a system without any data augmentation, the suggested speaker augmentation with SpecAugment method of [77] reduces word error rate (WER) by 30%, while reducing the WER by approximately 18% in comparison to a system with SpecAugment. When SpecAugment is combined, ASR benefits from this approach given that there is more real data available. Because texts are more easily

obtained from a variety of sources, such as the internet, additional texts from the Fisher corpus are used for speech synthesis, resulting in a lower WER. Table 4.10 and Table 4.11 demonstrate the results on 5h and 50h switchboard respectively.

Table 4.10: Low Resource Speaker Augment on 5 h SWB [77]

Switchboard					
Model	**Data**	**Aug**	**Virtual Spkr**	**SWBD**	**CH**
Baseline	5 h	TTS	0	35.6	48.2
	5 h	TTS + Spec Augment	0	27.2	39.9
Proposed	5 h	TTS + Spec Augment	25	27.1	40.3
	5 h	TTS + Spec Augment	100	26.8	40
	5 h	TTS + Spec Augment	300	**26.5**	**39.5**

Table 4.11: Low Resource Speaker Augment on 50 h SWB [77]

Switchboard					
Model	**Data**	**Aug**	**Virtual Spkr**	**SWBD**	**CH**
	50 h	None	-	25.6	39.2
	50 h	SpecAugment	-	20.2	32.5
	50 h	TTS	0	21.1	35.2
Baseline	50 h	TTS + SpecAugment	0	17.8	29.7
Proposed	50 h	TTS + SpecAugment	25	17.2	28.8
	50 h	TTS + SpecAugment	100	16.5	28.7
	50 h	TTS + SpecAugment	300	**16.5**	**28.2**

4.7 Synthetic Speech Augmentation Impersonation

4.7.1 Idea and Comparison

The assumption here (shared in our study), is that quality synthetic speech with varying prosody can be produced inexpensively. [75] uses the LibriSpeech dataset, augmented with synthetic speech. They train a very large end-to-end neural speech recognition model. In order to learn different speaker identities, the Tacotron-2 model from the OpenSeq2Seq toolkit [80] is used with Global Style Tokens (GST).

Using an encoder-decoder architecture, the Tacotron-2 model generates Mel spectrograms from input text. It was discovered that the amount of dropout in the T2-GST model can be used to alter the model. Lowering the dropout rate caused the audio to be faster. As a result, synthetic data quantity was increased, and a more evenly distributed dropout rate was adopted. The suggested model is an end-to-end neural network that constructs characters from logarithmic Mel-scale spectrograms as inputs. A deep CNN model based on Wav2Letter [81] is employed.

The model is composed of seventeen one-dimensional convolutional layers and two fully connected layers. The Wave2letter model preprocesses a speech signal by sampling the signal's raw audio waveform using a 20ms sliding window with a stride of 10ms. Afterwards, log-Mel filterbank energies of size 64 are generated from these frames and used as input features in the model. The model is trained using Connectionist Temporal Classification (CTC) loss, and outputs a sequence of letters that corresponds to the spoken input.

4.7.2 Improvements and Drawbacks

To determine WER improvement using synthetic augmentation, an experiment was carried out with 24, 34, 44, and 54 layers. As demonstrated in Table 4.12, models trained on both combines synthetic and natural datasets outperformed models trained on the original LibriSpeech dataset.

To determine the optimal sampling ratio for synthetic data and LibriSpeech, testing was conducted by training on LibriSpeech-only, a 50/50 split, a 33/66 split, and a pure synthetic dataset. A 34-layer model was used for all of the testing. Demonstrated in Table 4.13, results show that sampling natural and synthetic data at a 50/50 rate is most optimal.

Table 4.12: Greedy WER on LibriSpeech for Different Models and Datasets

Model	Data	Dev		Test	
		Clean	**Other**	**Clean**	**Other**
attention-Zeyer et al	LibriSpeech	4.87	14.37	4.87	15.39
w2lp-24	LibriSpeech	5.44	16.57	5.31	17.09
	Combined	5.12	16.25	5.16	17.01
w2lp-34	LibriSpeech	5.1	15.49	5.1	16.21
	Combined	4.6	14.98	4.66	15.47
w2lp-44	Combined	4.24	13.87	4.36	14.37
w2lp-54	Combined	4.32	13.74	4.32	14.08

Table 4.13: WER for Different Ratios Between Natural and Synthetic Datasets

Model	Data	Dev		Test	
		Clean	**Other**	**Clean**	**Other**
w2lp-34	Natural	5.1	15.49	5.1	16.21
w2lp-34	**50/50**	**4.6**	**14.98**	**4.66**	**15.47**
w2lp-34	33/66	4.91	15.18	4.81	15.81
w2lp-34	Synthetic	51.39	80.27	49.8	81.78

Using synthetic data effectively allows for the development of big neural speech recognition systems. However, to acquire the greatest results, synthetic data should be blended with natural data in the appropriate ratio.

4.8 Real-Time Zero-Shot Voice Style Transfer with Convolutional Network

4.8.1 Idea and Comparison

We study an interesting method that presents a neural network for zero-shot voice conversion (VC) that does not require any parallel or transcribed data. The method employs automated speech recognition (ASR) and speaker embedding models that have been pre-trained using data from a speaker verification task. Except for a modest pre-trained RNN for speaker encoding, the model is entirely convolutional and non-autoregressive. The proposed method, dubbed as “ConVoice” [6] has a convolutional architecture that allows it to convert speech of any length without sacrificing quality.

Parallel and non-parallel VC are the two basic methods to the voice style transfer challenge. The parallel VC technique makes use of a parallel speech dataset comprised of pairs of utterances from the source and target speakers. In practise, this is a complex, time-consuming, and costly structure to construct. A parallel corpus is not required for the non-parallel VC technique. Numerous non-parallel VC models have been influenced by image style transfer in computer vision concepts such as Generative Adversarial Networks (GANs) [82] and Variational Autoencoders (VAEs). However, GANs are challenging to train, whereas VAE’s outputs are frequently over-smoothed and do not create quality human-like speech. A non-parallel VC strategy based on the encoder-decoder paradigm is another non-parallel VC approach. The encoder extracts linguistic information from the source utterance using an ASR model, and the decoder synthesizes new speech in the target speaker's voice.

4.8.2 Method Description

This method is inextricably linked to the TTS Skins model [83]. TTS Skins is an encoder-decoder network that makes use of acoustic properties collected from the Wav2Letter ASR model. The primary distinctions between TTS Skins and this model are:

- A modern QuartzNet-5x5 encoder [84] is used that is 20 times smaller than Wav2Letter [81].
- A separate fully-convolutional decoder and pre-trained non-autoregressive WaveGlow vocoder [85] is used instead of WaveNet.
- A speaker encoder that has been pre-trained on a speaker verification task to extract speaker embeddings is used rather than a lookup table, enabling zero-shot conversion.

As a result, this model is substantially smaller, more computationally efficient, and requires no new data collection when switching to a new speaker's voice. The basis of Model Architecture is an encoder-decoder paradigm, and it is composed of four neural networks that are trained individually: audio encoder, speaker encoder, decoder, and vocoder. The audio encoder is pretrained on LibriTTS [86]. The speaker encoder is from the GitHub repository trained on VoxCeleb1 [87]. The Mean Opinion Score (MOS) was used to assess the naturalness and speaker similarity of synthetic speech on Amazon Mechanical Turk. The results were compared to those of N10 [25], the top system at VCC2018.

4.8.3 Improvements and Drawbacks

This work was a part of a 2018 challenge called Voice Conversion Challenge (VCC2018) [88]. The best model was called N10. ConVoice results are being compared to N10. Two models are proposed, one is zero-shot and other is the model fine tuned on the VCC2018 dataset [89]. A disadvantage of this strategy is necessity of using a pre-trained ASR model.

The evaluation is based on two criteria; speech naturalness and speaker similarity.

- Speech Naturalness: ConVoice performance is slightly worse than N10 in a zero-shot setting, as it generates some background noise. However, after fine-tuning the model using a small sample of data comprising utterances pronounced by target speakers, this unpleasant noise almost completely disappears and the quality of synthesised speech increases to equal that of N10.
- Speaker Similarity: Speech similarity of samples that are generated by ConVoice in a zero-shot setting is lower than that of the fine-tuned model.

This Page Intentionally Left Blank

Chapter 5

Speech Variations and Datasets

Speech interpretability degrades due to various effects of the surrounding environment even for the best recognizer - humans [90]. Taking a certain type of variability into account takes place by modifying or adapting one of the ASR system components, thus alleviating the effect of variability in real-life scenarios. Hence, adaptation to various noise sources is a must to avoid deterioration in recognition. There are many sources of interference with speech. Some of these sources are as follows, but not exclusive to:

5.1 Speaker Side

1. The different dialogues or accents of different speakers: This is one of the very important sources of speech distortion, since the phonemes and prosody of speech are different as well as the use of some words or expressions, can be different from the language modelling perspective [91].
2. The emotional state of the speaker: The speaker's emotions introduce certain prosodies to the speech signal as well as variability in the duration of utterances and perhaps some non-linguistic phones such as giggling or sniffing while speaking.
3. The physical state of the speaker: An ill speaker can have changes in his fundamental frequency, difficulty to utter same phonemes such as nasals in the case of the flu as an example.

5.2 Communication Channel

1. The surrounding environment [91],[92].
    - Existence of other speakers which is highly distorting to speech recognition.
    - Existence of structured noise such as machinery, music, etc.
2. Existence of reverberation, especially in closed environments such as cars and meeting rooms.
3. Existence of unstructured noise such as flicker noise due to electrical interference with the acquisition device, or errors in the storage or transmission processes.

Human perception can unconsciously deal with almost all of the above issues even if some of them are combined together, unlike ASR systems which must be trained to cope with such issues. Most of the research in literature targets a solution to one of the above issues, therefore when another source of distortion is introduced, ASR performance deteriorates significantly.

Due to the vastly growing scale of ASR applications, it is inevitable to deal with variations such as the above in speech and solve them. Users of ASR systems increase as the number of applications increase. Hence, ASR will not find public acceptance if there are so many constraints on using them, especially if these constraints are placed on the speaker and his or her dialect. A broad spectrum of users which have different accents and dialects will be frustrated towards commercial ASR systems that do not accommodate such differences.

The most practical approach towards tackling the different sources of speech variation is to train models of larger capacity on datasets containing all sorts of anticipated speech variations. There are two major downsides for this approach:

- Datasets containing all types of variations cannot be available to all ASR application, especially for the new applications, where there is yet not enough data or feedback from previous users.
- All available learning models have a limited capacity for learning, and cannot capture all such variations on their own without the introduction of hand engineered features. This can be seen from the performance of end-to-end LSTM ASR systems built using raw signals as in [93]

Therefore, this thesis will study first the amount of degradation in various ASR models when put to test vs. what it trained on. Next, it will study how the learning capacity of a model differs when introduced with some variability in the training phase. Finally, it will study how to synthesize such variability to train models that are ready to face real application settings before enough data can be collected.

There are several types of variability when it comes to speakers such as age, gender, accent, and health conditions. With the current datasets in our library, we can test the models' generalization, adaptation, and goodness-of-fit for age, gender, and accent. According to the description of the datasets, the training and testing data are organized to evaluate the aforementioned ASR model capabilities. The main issue with ASR is the discrepancy between the speakers in real-life and those involved in the training of the system. Data augmentation is a very powerful tool when it can simulate the actual variabilities in real-life. Nevertheless, the ASR model must be able to benefit from this added information.

5.3 Datasets used

1. **The Hispanic-English Dataset:** This dataset contains approximately 30 hours of English and Spanish conversational and read speech with transcripts (24 hours) and metadata collected from 22 non-native English speakers. Conversations are task-oriented; drawing on exercises similar to those used in English second language instructions, and designed to engage the speakers in collaborative, problem-solving activities [94].

2. **The Nationwide Speech Project (NSP):** The purpose of the NSP was to collect a large amount of speech produced by male and female speakers representing the primary regional varieties of American English: New England, Mid-Atlantic, North, Midland, South and West. This release contains approximately 60 hours of speech or nearly one hour of speech from each of 60 white American English speakers, including five male and five female speakers from the six dialect regions reading words and sentences [95].

3. **TIMIT:** TIMIT features broadband recordings of 630 speakers of eight main American English dialects reading ten phonetically rich sentences each. The TIMIT corpus contains orthographic, phonetic, and word transcriptions that are time-aligned, as well as 16-bit, 16kHz speech waveform files for each utterance [96].

4. **NTIMIT:** The NTIMIT dataset has been collected by transmitting all 6,300 original TIMIT recordings through a telephone handset and over various channels in a telephone network and re-digitizing them. The recordings were transmitted through ten Local Access and Transport Areas, half of which required the use of long-distance carriers. Passing the TIMIT speech data through this lossy network makes it suitable for testing variations in the communication channel [97].

This Page Left Blank Intentionally

Chapter 6

Model Capacity Effects in Modelling Synthetic Data for Environmental Variability

6.1 Introduction

Accuracy for Automatic Speech Recognition (ASR) systems often degrades due to various effects of the surrounding environment of the speaker, speaker variabilities, and the acquisition device or transmission channel [91], [92]. These types of variations can even affect the recognition of the best recognizer; humans [90]. Building ASR systems robust to all sorts of variabilities is an impossible task. Therefore, most literature targets one type of variability and proposes a solution to it. Taking a certain type of variability into account takes place by modifying or adapting one of the ASR system components, thus alleviating the effect of variability in real-life scenarios. Table 6.1 lists different components accounting for certain variabilities within an ASR system and corresponding exemplars of solutions in the literature.

Table 6.1: Noise Robustness in different ASR components

ASR component	Exemplar usage for robustness
Training data	Speech synthesis for data augmentation [98]
Pre-processing	Filtering and noise modelling [99]
Feature extraction	Noise robust features [100]
Acoustic/Language Modelling	Speaker adaptation [101], language domain adaptation [102]
Post-processing	Language post-processing [103], and classifier fusion [104]

In this chapter, we examine the ability of different ASR acoustic models in sustaining performance when trained with TIMIT [96] data and tested using the network distorted version of TIMIT, the NTIMIT [97]. In addition, the study covers the capacity of the ASR model to incorporate variability in its structure using different proportions of training data similar to the test environment. We also evaluate the robustness of the same models against varying noise levels using a distorted version of TIMIT testing data using the QUT-NOISE toolbox [105] and a clean training dataset.

The study in this chapter is organized as follows; section 6.2 discuses acoustic modelling options, section 6.3 describes the experimental setup, sections 6.3.1 and 6.3.2 report results and commentary respectively, and section 6.3.3 concludes the results.

6.2 Acoustic Modelling

The acoustic model is the main part in any ASR system; every other component just serves and assists the acoustic model. There have been many attempts to change the regular form of the ASR system to become an end-to-end neural network system [106]. In general, all acoustic models must account for variations in time, and this was evident in distance measures using Dynamic Time Warping (DTW) [107], stochastic or probability modelling using Hidden Markov Models (HMM) [108], or discriminative modelling using Conditional Random Fields (CRF) [109], and finally neural network modelling using Recurrent Neural Networks (RNNs) [110] with its variations.

Depending on the nature of the application, the ASR system can be adapted to perform better. The more generic the application and the speakers using the ASR are, the poorer the performance will be. Limiting the generality from any aspect improves the performance of the ASR. From the acoustic perspective, the limitation can be building a gender, a speaker, or an environment specific model. This type of limitation requires either building several models for each specific type or modelling a transformation model to normalize the variabilities amongst the speakers. Either way the system will require more data to model each of the specialized models to perform the ASR task.

The capacity of any machine learning model is proportional to its complexity. Nevertheless, increasing complexity usually means an increased number of learnable parameters, and hyper-parameters, which are parameters that need to be selected by the model designer and are not learned from the data, requiring a lot of data to properly learn. In this study, we are going to examine four virtues of different ASR acoustic models concerned with discrepancy between the training and testing environments:

1. Generalization when the testing environment is significantly different from the training environment.
2. Adaptation to learn from fractional exemplar data.
3. Specialized performance when there is no discrepancy between the training and testing environments.
4. Robustness against varying noise levels in the testing environment.

6.3 Synthetic Noise data Experiment

The datasets used for the experiments are the TIMIT, NTIMIT, and real noise distorted TIMIT using the QUT-NOISE dataset [103]. To assess the performance of the acoustic modelling only, the performance is measured by Phone Error Rate (PER) and not Word Error Rate (WER) to remove the effect of the language model on the system's performance. The models that are going to be studied are the baseline Gaussian Mixture Model (GMM)-HMM, the improved models using Maximum Likelihood Linear Transform (MLLT) and Linear Discriminant Analysis (LDA) [111], the hybrid HMM – Deep Neural Network (DNN-HMM) [9], and Subspace Gaussian Mixture Model (SGMM) [10]. Model training is conducted using KALDI [112]. The experiments aim at testing the four aforementioned virtues of each model as follows:

- **Generalization**: This test aims at exploiting the performance of the ASR system when the training and testing environments are different. In other words, the ASR will not be trained with data similar in nature to that of the testing data. For this experiment, the models are trained using the TIMIT training data and tested using the NTIMIT test data. The generalization power will be reflected in the amount of deterioration in the performance of each model relative to the ideal case where the training and testing data are from the same environment. The generalization is inversely proportional to the amount of deterioration in the model performance for varying levels of relevancy of the training data to the testing.
- **Adaptation**: This test aims at identifying how fast the model improves its performance when provided with varying fractions of training data that match the testing environment. This is done by replacing a fraction of the TIMIT training data with its NTIMIT counterpart. The rate of improvement of performance on the test data versus the fraction of training data introduced will be representative of the adaptation power of the model; the higher the rate the better the adaptation is.
- **Specialization**: This test specifies the model's ideal performance when training and testing data are from the same environment. For this experiment, the models are trained and tested using the NTIMIT dataset. Although this case might not be practical in real life, it is however an indicator of the model's best achievable performance, which is used as a reference for the other tests. of relevancy of the training data to the testing
- **Robustness**: This test uses the QUT-NOISE distorted TIMIT for the CAR and HOME scenarios at different Signal-to-Noise Ratio (SNR) values for the test dataset and clean training dataset.

6.3.1 Results

Figure 6.1 shows the rate of improvement of the different models graphically versus changing the percentage of similarity of training data to testing data. It can be directly noted that as the portion of the data induced from the NTIMIT into the TIMIT training data increases, the performance of all models improves but at different rates. All models experience rapid improvement when introduced with the first portion of dataset from the NTIMIT data and then come to saturation after the sixth portion. The data is laid out in Table 6.2.

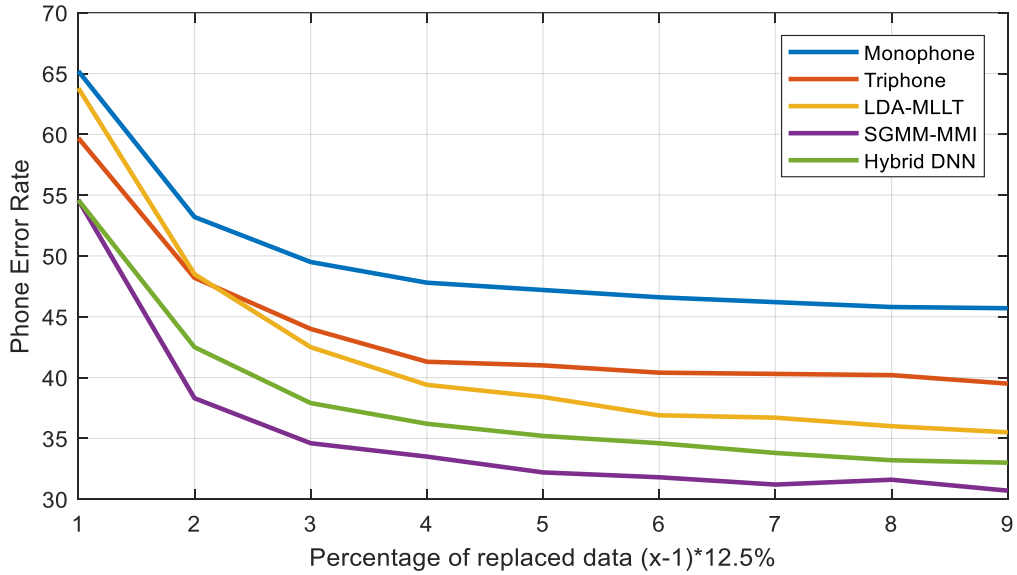


Figure 6.1: PER% vs different portions of perturbated training set as multiples of 12.5%

Table 6.2: PER% vs different portions of the perturbated training dataset

ASR Model	0%	12.5%	25%	37.5%	50%	62.5%	75%	87.5%	100%
Monophone	65.2	53.2	49.5	47.8	46.6	46.6	46.2	45.8	45.7
Triphone	59.7	48.2	44	41.3	40.4	40.4	40.3	40.2	39.5
LDA-MLLT	63.8	48.5	42.5	39.4	36.9	36.9	36.7	36	35.5
SGMM-MMI	54.6	38.3	34.6	33.5	31.8	31.8	31.2	31.6	30.7
Hybrid DNN	54.6	42.5	37.9	36.2	34.6	34.6	33.8	33.2	33

Figure 6.2 and Figure 6.3 show the improvement of PER versus the variation of SNR for the HOME and CAR scenarios. It is very interesting that the general improvement trends for different models with increasing the SNR are very similar to that of the experiment laid out in Figure 6.1. The results are tabulated in Table 6.3 and Table 6.4.

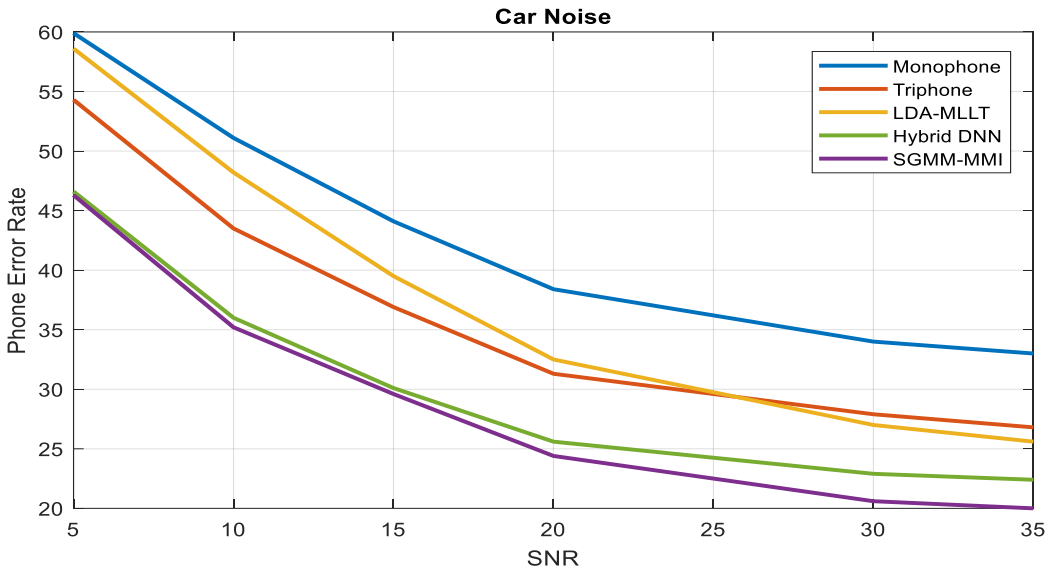


Figure 6.2: PER% vs different SNR values for noise distorted TIMIT set in CAR scenario

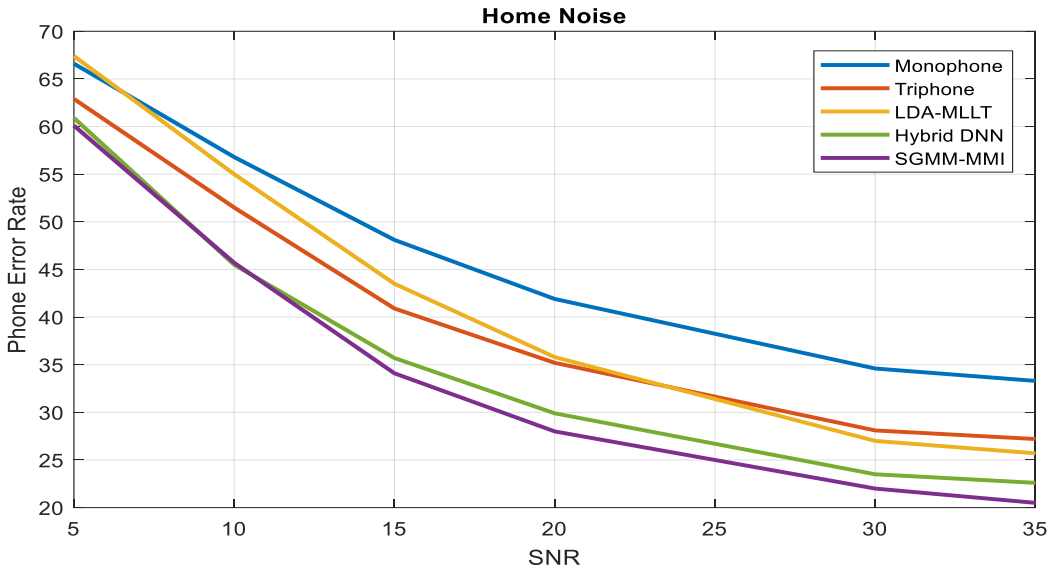


Figure 6.3: PER% vs different SNR values for noise distorted TIMIT set in HOME scenario

Table 6.3: PER% for varying SNR in the CAR noise environment

Model\SNR	5	10	15	20	30	35
Monophone	59.9	51.1	44.1	38.4	34	33
Triphone	54.3	43.5	36.9	31.3	27.9	26.8
LDA-MLLT	58.6	48.2	39.5	32.5	27	25.6
SGMM	49.5	38.5	31.6	26.3	23.4	22.2
Hybrid DNN	59.4	47.5	38.6	31.3	26.8	25.3

Table 6.4: PER% for varying SNR in the HOME environment

Model\SNR	5	10	15	20	30	35
Monophone	66.6	56.8	48.1	41.9	34.6	33.3
Triphone	62.9	51.5	40.9	35.2	28.1	27.2
LDA-MLLT	67.4	55	43.5	35.8	27	25.7
SGMM	62.2	48.2	36.3	31.2	23.7	22.9
Hybrid DNN	66.9	55.1	43.3	35.5	27.2	25.9

6.3.2 Comments on Results

There are several conclusions and recommendations that can be taken out of these test results:

- The difference in performance between the models is not significant when there is dissimilarity between the training and testing environments. Also, adding transformations such as LDA and MLLT can actually harm the performance under this scenario. The reason is that both transformations rely on a learned linear mapping of the features that separate the data of each class as much as possible. Hence, if there is a mismatch between the training and testing environments then the learned transformation will be misleading to the classifier causing a decrease in performance.

- The improvements in both the triphone and monophone models are consistent, which means that both of these models have similar adaptation power.
- Although the hybrid DNN-HMM model starts with PER equivalent to that of the SGMM, the SGMM adapts a lot better. What makes a model capable of fast adaptation is its balance between bias and variance. DNN is mostly famous of for its large variance, while the regular full GMM-HMM has also high variance. The SGMM manages to tie parameters of the learning model in a way that reaches a better balance than the DNN.
- The best generalization and specialization model is the SGMM-MMI model.
- The best adaptation is HMM with LDA and MLLT transformations.

All of the above comments also apply to the noise varying experiment that assess the conclusions. Hence, the performance of models with fractional data similar to that of the test environment is similar in improvement trend to varying the SNR of the test data. This shows that the models' virtues are consistent when handling discrepancies between the training and testing environments.

6.3.3 Conclusion

This study makes a deeper investigation in the performance of ASR models under real-life scenarios facing ASR systems nowadays. We study four main virtues of some of the commonly used ASR systems; generalization, adaptation, specialization, and robustness. The generalization power of different models can heavily impact the performance of the model in new applications with limited training data in the new working environment. Adaptation is an indicator of how well the model improves as new data is added in the training set. Models with transformation showed the best adaptation power.

Classical monophone HMM-GMM models adapt similarly with more data as the more complex triphone models. Hence, if the initial performance is acceptable, the simpler model is a better choice since it will improve in a similar manner to the more complex triphone model. As mentioned before and as most literature research is conducted, benchmark datasets have a certain nature regarding training and testing data. Such restrictions force the learning models to a certain type of speech, and hence do not assess the performance of the models for real scenarios where they will be faced with a lot of variability. After identifying the capacities of different models in the thesis, the promising models will be provided with synthetically enriched datasets, and the improvement in performance will be evaluated.

The synthesized speech variations will target the most important and common variations in terms of user usage. Problems that can be mostly solved using extra hardware such as background noise elimination using a microphone array, or filters for electronic noise will not be considered due to current satisfactory user experience. Other speech distortions resulting from the speaker are the main concern since there are no hardware solutions for the problem. The speaker variations to be considered for synthesis are the variations in accent, native language, age, and the emotional state. The first two types are essentially of the same nature, where some phonemes are not uttered properly by the speaker. On the other hand, the emotional state of the speaker introduces some variations in the duration, intonation, and prosody of the speech signal.

6.4 Real Life Noisy Data Experiment with Augmentation

This experiment was conducted as part of a study to upgrade an existing model that is currently in use by an enterprise that provides a suite of situational awareness tools for police and first responders. The ASR model is one of the situational awareness tools that plays a critical role in providing real-time critical radio-communication transcriptions for police and first responders. This takes a huge load off law enforcement's dispatch and different departments attempting to relate critical data to each other when every second counts. Every emergency responder is able to view in real time, audio transcription of all fellow colleagues at or near the scene of a situation. It is without a doubt, a game changer.

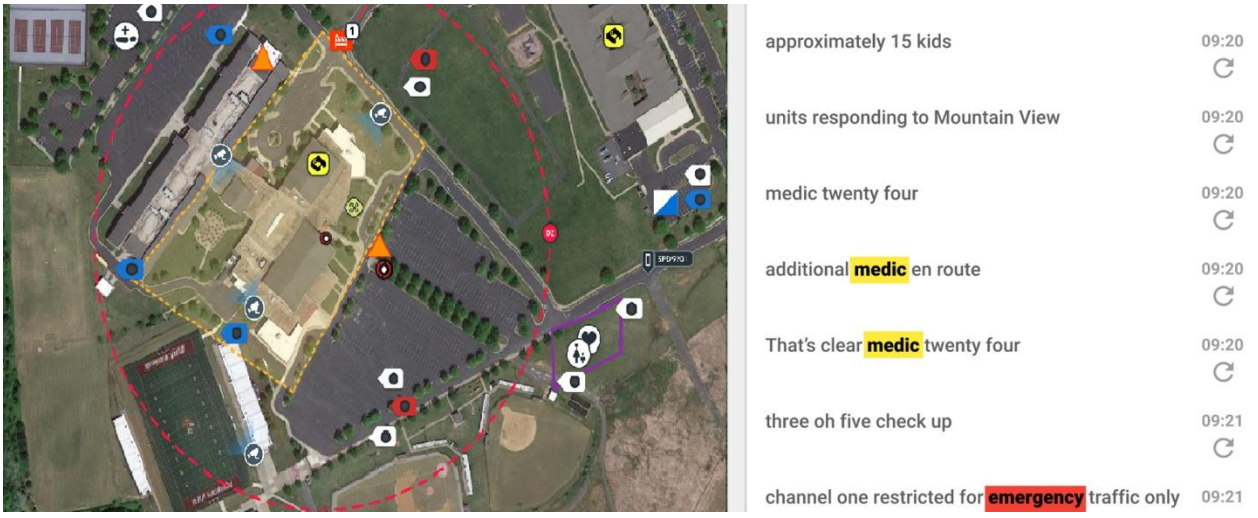


Figure 6.4: Situation Awareness Tool

6.4.1 Challenges

Although no proper documentation about the current deployed ASR model was communicated to us, we do know however, the current model is old with a WER that reaches beyond 25% at times. To start with, we were given 40 hours' worth of Radio-Traffic data, and at a later stage, we were handed approximately 120-hour dataset. The average duration per audio file is approximately 2.8 seconds. All audio data is transcribed in-house, and comes with many challenges:

1. Dataset is choppy, meaning the majority of the audios start after the speaker has begun speaking and end before the speaker has finished. The transmitted speech is missing some sounds at both ends of the transmission usually due to user error. The push-to-talk (PTT) button should be held down long enough before the start of speech and released once speech is complete. However, these are emergency situations, and the norm does not apply.
2. The data is noisy, as one can only imagine the different scenarios, such as fire, sirens, gunfire, firemen speaking under a protective mask while breathing heavily from an oxygen tank, and much more.
3. Speaker's information is unknown. These models are deployed per county. Each county has hundreds, if not thousands of emergency-personnel with daily radio transmissions, and with a wide variety of different accents.
4. Some audio files are automatically generated alerts (Text-To-Speech) with single tone frequency beeps of unique range. There is no reference to these beeps in the transcriptions. We will have to create a script to search for all these audios, and manually enter a reference for each beep in the transcriptions.

6.4.2 Experiment-A (40 hr dataset)

Experiment-A was conducted on the 40 hrs dataset, and since the nature of the data is that of noisy and choppy audio, we decided to enrich the dataset with augmented data as follows:

a) Speed perturbation factors: 0.9, 1, and 1.1 [113], [114]
b) Volume perturbation at factors: 0.125, and 2 [113], [114]

This will enrich the dataset to 120 hrs. When training the Monophone and Triphone models, we conducted three experiments to see what hyper-parameter values are most suitable for this dataset. We kept the

default values of the number of gaussians (2000) for Monophone training through all three experiments, but we selected a minimum, middle, and maximum hyper-parameter values for Triphone training. The suggested range of hyper-parameters for Triphone training according to Kaldi is 2,000 to 5,000 for the number of leaves, and 10,000 to 50,000 for the number of total gaussians. Our choice of hyper-parameter values was as follows:

1. In the first experiment, we fixed the hyper-parameters at the minimum value of 2,000 leaves and 10,000 gaussians.
2. The second experiment, we chose the hyper-parameter values to be 3,500 leaves and 30,000 gaussians.
3. For the third experiment, we chose the maximum hyper-parameter values of 5,000 leaves and 50,000 total gaussians.

6.4.2.1 Results

We establish from the experiments that a medium range of hyper-parameter values is most optimal as demonstrated by the WER in Table 6.5. Since medium range hyper-parameters showed the most promising results (38.77% WER) for the Triphone model, the alignments of the training set generated by the Triphone model are used for neural network training. The semi-final experiment is a hybrid 1D CNN-HMM model without any data augmentation producing a WER% of 24.2, while for the final experiment, augmented data is added to the model, thus showing a significant reduction in WER down to 17.03%.

Table 6.5: Radio-Traffic 40 hr dataset, 39 hrs training and 1 hr testing (WER%)

Model	Parameters		
	Low	**Medium**	**High**
Monophone	71.88	66.47	67.3
Triphone	46.43	38.77	40.41
Hybrid HMM-TDNN (No Augmentation)	-	24.2	-
Hybrid HMM-TDNN + Augmentation	-	**17.03**	-

6.4.3 Experiment-B (120-hr dataset)

Experiment-B was implemented on the 120-hr dataset, which comes with the challenges previously mentioned. For our Triphone model, we used the medium range hyper-parameter values used in experiment-A since they were the most suitable for this kind of dataset. We decided to train two models, one with speed and volume perturbation, and one with only speed perturbation so as to see what kind of impact would volume perturbation have.

6.4.3.1 Results

As demonstrated in Table 6.6, there is a slight difference between the model with volume perturbation and the one without.

Table 6.6: Radio-Traffic 120-hr dataset, 117 hrs training and 3 hrs testing (WER%)

Model	Test
Monophone	74.66
Triphone	48.73
Hybrid HMM-TDNN + Speed Perturbation Aug.	20.73
Hybrid HMM-TDNN + Speed and Volume Perturbation Aug.	**20.27**

6.4.4 Future work

Due to the nature of this dataset, we intend to enrich the dataset by synthesizing a clean dataset, mimicking radio communications. This can be implemented by adding reverberation, random choppings, random beep alerts, and background noise.

This Page Intentionally Left Blank

Chapter 7

Neural Style Transfer Data Augmentation

7.1 Introduction

Different types of speech distortions need to be synthesized so as to obtain a rich training dataset encompassing enough speech variability. Different synthesis approaches are proposed, and will be compared in terms of performance improvement of ASR systems trained using the enriched datasets.

A successful new trend in computer vision is concerned with what is termed artistic style transfer [36]. The main target of this trend is to transfer a given normal image into another image having the same content, but with a different drawing style, like brush coloring or mosaic for example. This approach mainly relies on Convolutional Neural Networks (CNNs) [115].

Inspired by the neural style transfer approaches for images, a synthetic variability-generating approach is proposed for the phonemes using style transfer of speech signals. This variability comes along with modeling distortions, since the modeling process is not ideal. Figure 7.1 summarizes the idea for image style transfer. The core idea is to capture the features of each accent (non-native) using a discriminating CNN, and then use the difference between the features extracted by the CNN and those resulting from the current image as a loss function. The approach assumes that there is an audio signal and just its accent or emotion will be altered.

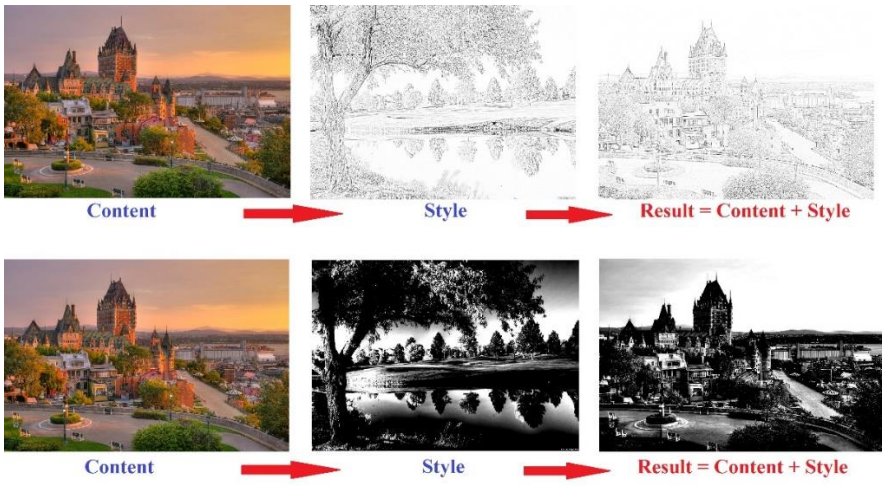


Figure 7.1: Neural Style Transfer of Images

While deep neural networks have demonstrated record-breaking performance on complex machine learning tasks in recent years, sometimes exceeding human performance levels, the majority of deep models heavily rely on large amounts of annotated data for individual tasks, which is frequently prohibitively expensive to obtain. A popular technique is to enrich smaller datasets by using label-preserving modifications to generate additional training examples from existing ones.

We offer style augmentation, a novel technique for data augmentation based on random style transfer, with the goal of increasing the resilience of Convolutional Neural Networks (CNNs) for both classification and regression applications. The style transfer can alter speed, prosody, and accent of a given audio file. This is performed by adopting an arbitrary style transfer network to execute style randomization, instead of calculating target style embeddings from a style audio, along with conventional classification tests.

We examine the influence of style augmentation (and data augmentation in general) on domain transfer tasks. We demonstrate that data augmentation considerably enhances resilience to domain shift and may be used in place of domain adaptation as a simple, domain neutral alternative. We demonstrate that when style augmentation is compared against a combination of other existing augmentation approaches, it can be easily integrated with them to increase network performance. We demonstrate our technique's effectiveness via domain transfer tests in enhancing the performance of various benchmark speech recognition systems.

By explicitly teaching variance for feasible changes that maintain semantic information, data augmentation transfers previous knowledge to a model. This is accomplished by applying the transform to the original training data, therefore yielding new samples with known labels. For instance, speed perturbation and pitch variation are widely used approaches for data augmentation. While these techniques are effective in teaching a model different speaking-ways, it does not capture complex variations such as accents and intonations.

Neural style transfer enables the distribution of low-level speech elements in an audio to be altered while maintaining semantic speech structure. Using this approach, we present Speech Style Augmentation (SSA), a technique for using style transfer to map the accents of arbitrary training audios while keeping their structure to another accent. While the original style transfer technique required a target style picture, our method uses a style audio which is essentially any target audio file with a single speaker converted to its MFCC domain as well as the x-vector information which convey the speaker identification information.

We assess our technique on a variety of domain adaption tasks in addition to typical classification criteria. To our knowledge, this is the first time, domain adaptation has been evaluated via data augmentation using Neural Style Transfer in speech. Normally, data augmentation is intended to mitigate overfitting and increase generalization to previously unseen speech in the same domain, but we argue that domain bias is a kind of overfitting and therefore should benefit from the same countermeasures.

While data augmentation is not domain adaptation, it may help lessen the requirement for domain adaptation by first training a more generic and robust model. While this strategy may not outperform domain adaptation to a given target domain in terms of performance, it has the benefit of boosting accuracy on all prospective target domains before they are observed, and without needing different processes for each.

In summary, this study investigates the idea of augmenting data using style transfer in order to train more robust models that generalize more successfully to data from unknown domains. Thus, our key contributions are as follows:

1- We offer a unique and successful strategy for transferring the audio of a certain accent and domain using style transfer network to another.
2- We examine the efficiency of utilizing style augmentation to indirectly increase performance on domain transfer tasks, which often entails post-training adaptation of a model to a particular target domain.
3- These contributions are bolstered by extensive testing on a variety of tasks and model architectures, aided by hyperparameter grid searches.

7.2 Related Work

7.2.1 Domain-Specific Bias

Domain bias or domain shift has long been a source of contention for academics involved in the training of discriminative, predictive, and generative models [116][117]. In summary, the issue is that a typical model trained on a given distribution of data from a single area would not generalize well to other datasets not encountered during training. For example, even though the job is the same and the training dataset is vast, an ASR trained on a native speaker dataset will perform less on non-native speakers even though they might be working in the same domain, such as news, for example.

Transfer learning is a common approach to the domain shift issue, in which a network is pre-trained on a comparable task using a big dataset and then fine-tuned on the new data [118]. This helps to mitigate the danger of overfitting to the source domain, since DNN-HMM features learnt on bigger datasets are more universal. Transfer learning, on the other hand, involves utilizing the same architecture as the pre-trained network and judicious use of layer freezing and early pausing [119] to avoid past information from being lost during fine-tuning.

Domain adaptation is another strategy for tackling domain shift [120][121]. It comprises a number of strategies for adjusting a model post-training to increase its accuracy on a given target domain. This is often achieved by some method of decreasing the distance between the source and target feature distributions. Certain techniques have been presented to reduce the Maximum Mean Discrepancy (MMD) [122], which is a measure of the distance across domains, while others have employed adversarial training to discover a representation that minimizes the domain discrepancy without sacrificing source accuracy.

While many adversarial domain adaptation strategies concentrate on discriminative models, domain transfer has also been used in research on generative tasks [123][124]. To address the disparity between the two domains, adaptive batch normalization is presented. This is More pertinent to our study, which makes use of picture style transfer to conduct domain adaptation.

Although it may deliver spectacular results, and is often useful, Domain adaptation is, however, restricted in that it can only aid in the generalization of a model to a single target domain. By contrast, our technique brings more variation into the source domain via data augmentation, which may improve the model's overall resilience, resulting in improved generalization to a large number of possible target domains without needing data from them.

Data augmentation has been a widely used strategy for enhancing the generalization of DNNs. Data augmentation inflates a dataset artificially by deriving additional samples from the originals using label-preserving transformations. Because data augmentation is a technique for directly teaching invariance to the transform being employed, any transform that mimics intra-class variation is a possibility.

7.3 Proposed Approach

We employ data augmentation using a typical Neural Style transfer model where there is an observation and a style observation that we want to morph the observation to. Of course, the original style transfer worked with images [125], so to employ the same strategy we had to change the structure to 1D CNN

instead of 2D, which is a straightforward process. The inputs however differ; we take 3 inputs, the original wave, the original MFCCs, and the style wave MFCCs.

The output is the style transferred MFCC. So, we directly create an observation with its MFCCs extracted to be used directly in the training process. We also found that utilizing a style from a speaker close in characteristics to that of the original make the results better. Hence, we first identify pools of speakers using the x-vectors [126], and then map from each pool of speakers to the other.

7.4 Experimental Setup

We use the TIMIT dataset to augment the training dataset of Hispanic-English with various proportions. To assess the performance of the acoustic modelling only, the performance is measured by Phone Error Rate (PER) and not Word Error Rate (WER) to remove the effect of the language model on the system's performance.

The models that are going to be studied are the baseline Gaussian Mixture Model (GMM)-HMM, the improved models using Maximum Likelihood Linear Transform (MLLT) and Linear Discriminant Analysis (LDA) [111], the hybrid HMM – Deep Neural Network (DNN-HMM) [9], and LSTM-HMM [10]. Model training is conducted using KALDI [112].

7.5 Results

Table 7.1 shows the results of the neural style data augmentation. The synthesized augmented data relative to the original is shown as well as the corresponding PER for the different models. It can be readily seen that the LSTM-HMM shows the best improvement when more synthetic data is added. That can be due to its controlled capacity relative to the DNN, which might be still striving for more data, and the classical GMM-HMM, which is an underfit for the different variations.

Also, it is notable that all models tend to show a decrease in performance when the synthetic data portion goes beyond 50%. This can be due to an adverse effect of distortions in the training data relative to the original data, and the model starts to focus on the distortions rather than generalizing.

Table 7.1: PER results of NST Data Augmentation on Hispanic-English test results

Exp	Training Dataset (Percentage)	Model	PER
1	Hispanic-English	GMM-HMM-si	47.3
2		GMM-HMM	40.1
3		DNN-HMM	32.6
4		LSTM-HMM	31.8
5	Hispanic + SynHispanic25%	GMM-HMM-si	47.2
6		GMM-HMM	39.7
7		DNN-HMM	28.1
8		LSTM-HMM	26.7
9	Hispanic + SynHispanic50%	GMM-HMM-si	46.8
10		GMM-HMM	39.3
11		DNN-HMM	27.5
12		LSTM-HMM	**25.8**
13	Hispanic + SynHispanic75%	GMM-HMM-si	47.1
14		GMM-HMM	39.6
15		DNN-HMM	28.0
16		LSTM-HMM	26.1
17	Hispanic + SynHispanic100%	GMM-HMM-si	47.4
18		GMM-HMM	40.3
19		DNN-HMM	32.1
20		LSTM-HMM	28.2

7.6 Conclusion

The neural style transfer is a versatile technique to generate more data for speech recognition. It can manipulate accents and improve the performance of different ASR systems. However, its computational requirements are immense due to the convolutional operations and the calculation of the gram matrix. It can also be distorted if there are significant mismatches between the original and style speakers.

Chapter 8

WaveNet Generating Model

This approach was successful for synthesizing speech from text. This can make the generation process of synthetic speech easier since it only requires text [127]. Nevertheless, this approach might require more data to produce robust results. This will be one of the comparison points when comparing the different approaches. Another merit for this approach is that it is a single model that can be trained on all types of variability, and simply conditioned on one or more of these types during generation. This model is also based on CNNs, but its convolutional layers do not convolve contiguous windows of samples, but rather convolve every other sample to capture long term dependencies with lesser computations. It is referred to as dilated convolution.

Data augmentation is critical for ASR systems to prevent overfitting and boost generalization. Previous research in ASR data augmentation provided numerous methods by conducting speed perturbation or spectrum transformation. Because data enhanced in this way has comparable acoustic representations as the original data, it has minimal use in terms of strengthening the acoustic model's generalization. We propose a voice conversion strategy employing a generative model (WaveNet), which creates a new utterance by transforming an utterance to a specific target accent, in order to avoid creating data with low variety. By reducing the usage of acoustic characteristics, our technique synthesizes speech with a variety of pitch patterns. Using the Hispanic-English dataset, we show that our approach outperforms conventional data augmentation strategies like speed.

8.1 Introduction

Large training datasets with rich patterns became increasingly important as the capacity of ASR systems grow. Building a labelled dataset, on the other hand, is costly and time-consuming, not to mention privacy issues that are now hindering the acquisition of data for new applications. Data augmentation (DA), which enhances the amount of training data by adding altered samples that keep the original labels, is a frequent technique for dealing with this problem. Speed perturbation is one of the most frequent DA techniques in ASR. However, since the acoustic properties of the speech data created by these approaches are similar to those of the original, they have little variation. As a result, we present a unique strategy for increasing variety in synthetic data using a generative model.

In this study, we present WaveNet-DA, a data augmentation technique based on the generative model WaveNet [40]. The following are some of the benefits of our method.

1. First, we use a generative model in our method. Previous approaches have focused on feature changes. The disadvantage of feature modification however, is that the acoustic model (AM) may be able to tell the difference between the original and altered data. The enhanced data is no longer a novel representation once the acoustic model understands the connection, and thus its value is diminished.
2. Second, instead of using traditional vocoders, we use WaveNet to produce utterances. During parameterization, traditional vocoders lose detail information, resulting in artefacts in the output speech. These artefacts, which are not present in actual data, might make it difficult for the AM to generalize on it.
3. Finally, our method produces speech with a variety of pitch patterns. Vocoder parameters like fundamental frequency and spectral information have no effect on our WaveNet-DA. As a result, since WaveNet is not provided with specifics of acoustic properties such as pitch variability, the synthetic speech might have a variety of pitch patterns. We demonstrate that the suggested approach outperforms DA utilising speed perturbation on the Hispanic-English corpus.

8.2 Relation to Previous Work

Synthetic data training is now extensively used for a variety of applications due to advancements in deep generative models. However, most research works in speech recognition have presented strategies that primarily involve the change of existing acoustic data, such as speed perturbation and vocal tract length perturbation. Also, a DA approach based on a variational autoencoder (VAE) [128] was proposed, and shown that it enhances an AM. We again provide a DA with a generative model, but this time using WaveNet, which is more suitable to synthesizing waveform data.

WaveNet, which was previously suggested as part of a TTS system with linguistic conditioning elements, has been re-imagined as a standalone system. Recently, it has been examined as a voice conversion vocoder with acoustic properties. Our ultimate objective, unlike previous research, is to supplement data for AM training. For this goal, we synthesize speech with a variety of pitch patterns by ignoring specific acoustic information, and we demonstrate that doing so enhances ASR performance.

8.3 The WaveNet

WaveNet is an autoregressive network that directly predicts a raw waveform sample-by-sample. It was suggested as a generative model for creating high-quality signals. The model approximates the joint probability of a signal for the input sequence by a receptive field of length *R*. Its architectural design is mostly made up of stacks of residual blocks, such as 2x1 dilated causal convolution, gated activation, and 1 x 1 convolution [40].

The WaveNet used in this work processes windowed speech waveforms with 64-stacks of residual layers [40]. Features are extracted every 25 ms resulting in a feature frequency rate of 100 Hz. It uses a transposed convolution layer to upsample to the necessary frequency (16 kHz), which is then followed by 1-D convolution. Finally, using the outputs from the conditioning network and the speaker embedding vector, the waveform synthesis component creates audio at 16 kHz.

8.4 Experimental Setup

We follow the original WaveNet architecture which employed 64-stacks of residual layers with dilated convolutions of size 25 ms and cross entropy loss to train DNN-HMM AM. Triphones with forced alignments were used to train it using GMM-HMM AM that was previously trained. Its input feature is a filter bank with 40 Mel-scale coefficients plus energy value and first and second temporal derivatives, resulting in a 123-dimensional input vector each frame. We employed a 25 msec long hamming window every 10 msec for the filter-bank feature, and each input feature was normalized for each unique syllable. A pruned trigram language model was utilized with a beam-size of 10 during decoding.

Stochastic gradient descent (SGD) is used to update model parameters in all circumstances, with a fixed learning rate of 1e-3 and a mini-batch size of 16 observations. For the baseline system, 30 training epochs were employed, whereas a smaller number of epochs were used for the advanced system. To maintain the enhanced system's training period comparable to the baseline system, 5 epochs were employed

8.5 WaveNet-DA Design

To train the WaveNet-DA system, we used Kaldi [129] to train a GMM-HMM system on the training set. We retrieved forced-alignments from the GMM-HMM AM's 3392 triphones to prepare data for DNN-HMM

AM. The alignment was then transformed to phoneme sequences and used to train the WaveNet model. We employed phoneme context data as input to the stacks, which included two preceding and two subsequent phonemes for each phoneme. During the procedure, log-energy values, another local auxiliary characteristic, were also acquired in the form of a one-dimensional sequence. Finally, we use a trainable embedding vector to represent speaker information as the global condition.

The following are the WaveNet model's hyper-parameters: There are 30 layers, 64 residual channels, 256 skip channels, and 512 dilation lengths. A speaker embedding dimension of 16 was also specified. Dropout was applied to all three residual stacks in a conditioning network, as well as the linear transformation layer on the speaker, with a probability of 0.15.

8.6 Perturbation of Speed

For different sizes of corpora, the speed-perturbation strategy is believed to be the most effective augmentation method. Using the Sox audio manipulation program [130], two speed-perturbed duplicates of the original training data were created by adjusting the speed to 90% and 110% of the original speed. We used the Kaldi GMM-HMM method to build alignments for this 3-fold training set.

8.7 Results

WaveNet-DA shows higher improvement for all ASR models as displayed in Table 8.1 with data added that comprises 50% of the Hispanic-English set, yet of course the amount of increase differs from one model to the other dependent on its capacity. The more the model has capacity the more it improves. It might seem unexpected to have the DNN score lesser performance than LSTM, but because LSTM has a more constrained way of tying the weights and builds relationships between the observations in time, it benefits more than the fully connected DNN.

Table 8.1: Results for WaveNet Augmentation vs. Speed Perturbation

Augmentation	Model	PER
Hispanic + SynWaveNet	GMM-HMM-si	41.3
	GMM-HMM	34.2
	DNN-HMM	23.3
	LSTM-HMM	**21.9**
Hispanic + Speed Perturbation	GMM-HMM-si	42
	GMM-HMM	37.1
	DNN-HMM	29.3
	LSTM-HMM	28.4

8.8 Conclusion and Futur work

We propose and illustrate the efficiency of a WaveNet-DA methodology without vocoder settings as a DA method for an ASR problem in this research. Unlike previous VC scenarios, our WaveNet creates speech resembling the target speaker with variety since it is locally conditioned exclusively on linguistic and energy information. It has shown that it improves the PER of the Hispanic-English using synthesized speech from the TIMIT corpus more than speed perturbation, which is a common and well known DA approach. To increase ASR performance further, we want to investigate the efficacy of our suggested strategies when applied to a bigger corpus.

This Page Left Blank Intentionally

Chapter 9

Recurrent Autoencoder

Autoencoder (AE) neural networks [131] are mainly used as unsupervised non-linear feature extractors. They are trained to regenerate the input at the output but with restrictions on the size of the hidden layers. A famous application of such AE is de-noising, and was applied to speech de-noising as in [132]. This approach trains an AE on clean speech, and at testing time, the AE is introduced with noisy data and clean speech is expected at the output.

Inspired by this approach, a recurrent AE (RAE) approach is proposed to be trained on non-native or accented speech and conditioned on a certain phoneme using a multiple input network [127]. The reason for conditioning is that some non-native accents mistakenly pronounce some phonemes with others. If the network is left freely to encode native speech to its non-native counterpart, the network will not consider such flaws in pronunciation and simply choose the phoneme closest to the native.

The same procedure can be followed for introducing different physical states for the speaker. Nevertheless, such data is not widely available and is not studied specifically [91]. Hence, a spectral manipulation approach will be followed to introduce the required distortions according to a set of studied effects of illness [133]. Finally, for the emotional state synthesis, another set of manipulations will be introduced according to previous studies [134].

9.1 Introduction

The approach of data augmentation and feature extraction for acoustic modelling is explained utilizing a Recurrent Variational Autoencoder (RVAE). A RVAE is a generative model that is built using a deep learning framework and is based on variational Bayesian learning. It may produce new information by extracting latent values from its input variables. RVAEs are commonly utilized to create visual and aural output. In this work, a RVAE is used to enhance speech corpora and extract feature vectors from speech for acoustic modelling. To begin, the size of a speech corpus is augmented via the use of a RVAE framework to encode latent variables retrieved from original utterances. The latent variables retrieved from speech waveforms represent the waveforms' latent "meanings." As a result, latent variables may be employed as acoustic characteristics in ASR.

This study demonstrates the usefulness of data augmentation utilizing a RVAE framework and the utility of latent variable-based features in ASR. Acoustic modelling needs a speech corpus (training audio data) with phoneme (or syllable) transcription for ASR. As the number of training epochs rises, ASR performance improves. However, preparing a speech corpus containing phoneme labels for training acoustic models for particular applications, such as speech recognition for seniors and ASR with low-resource languages, is challenging. In such instances, sufficient training epochs for acoustic modelling cannot be created, and consequently enough ASR performance cannot be acquired.

As a result, many data augmentation approaches have been developed to expand the quantity of training data by artificially producing synthetic acoustic feature vectors. The majority of works on ASR data augmentation have presented ways for directly generating acoustic feature vectors, since speech waveforms are not required for ASR.

The initial purpose of this research is to enhance an acoustic model via the use of a generative model based on a deep learning framework to increase a speech corpus. This technique creates simulated speech waveforms without the use of speech synthesis techniques. VAE and its derivative models may be used to perform a variety of tasks, including picture and caption production. Using a recurrent-based deep neural network, a deep learning framework can readily handle temporal sequence signals (DNN).

The focus of the study is on the production of speech waveforms and feature extraction for acoustic modelling utilising a RVAE, a basic generative model implemented in a deep learning framework. A RVAE is trained with identical input and target vectors. Unsupervised training may be used to train a RVAE. A RVAE, on the other hand, may utilise additional instructor labels.

A RVAE is made up of two components: an encoder that extracts a latent vector from input (observation) variables and a decoder that reconstructs the original variables from the latent vector. A RVAE's fundamental construction is essentially identical to that of a standard bottleneck AE. However, the distinction between two distinct kinds of AEs can be made only at their bottleneck levels.

The encoder's network parameters are trained with the constraint that the values of latent vectors follow a Gaussian distribution. Assuming Gaussian distribution for the latent variables, the encoder predicts the mean and variance of the Gaussian distribution for the input variables. A latent vector may be created by sampling latent variables with Gaussian distributions and having the encoder compute the mean and variance values.

A latent vector contains "meanings," which are abstract representations of the input variables, from which a decoder may reconstruct the input variables. This work provides a strategy for augmenting data using a RVAE. A speech waveform is reconstructed by encoding a latent vector from the input waveform. The produced waveform has the same content and duration as the input waveform; nevertheless, speaker uniqueness varies greatly. Produced audio are then added to a training corpus for acoustic modelling.

In contrast to a conventional AE, a RVAE is not an ideal encoder-decoder model. A perfect model is capable of reconstructing a speech waveform that is identical to the original. Thus, the suggested approach converts an input signal to a latent vector (i.e., a meaning representation). The latent vector performance would idealy represent the style of the speaker not just store a compressed form of the waveform. Assuming that the RVAE-reconstructed speech waveform lacks some of the distinctive characteristics of the original speech, the reconstructed speech waveform is regarded as if it were delivered by another speaker. As a result, using a RVAE to generate speech waveforms may significantly expand the number of audios available for acoustic modelling.

The following are the study's primary contributions. To begin, it is shown that autonomously produced synthetic speech may be used to enhance the data in a voice corpus for acoustic modelling. Notably, employing a RVAE to supplement voice corpus data is a novel challenge. Second, the experimental demonstration of the availability of a latent vector recovered by a RVAE for acoustic modelling is made. The suggested latent-based feature is a novel acoustic characteristic that has not been published before, to our knowledge. This work demonstrates the resilience of a latent-based feature for acoustic modelling using low-quality speech data. The following are the benefits of the suggested method:

- A RVAE may be trained using an unsupervised training framework, which eliminates the need for phoneme labels.
- The proposed method does not require speaker statistics such as vocal tract length perturbation.
- By resampling a latent vector, many audio features are generated. As a result, a speech corpus may be readily enlarged.

On the other hand, although high-quality features may be recovered from the original, the utterances and durations of the reconstructed features might not be identical to those of the original. However, the RVAE cannot create several styles of a single audio waveform. The second purpose of this work is to determine if latent vectors generated from speech can be efficiently employed as acoustic feature vectors for

automatic speech recognition. Assuming that latent vectors contain meaning (phoneme features), they may be beneficial for ASR.

9.2 Proposed Approach

To minimize the discrepancy between the input and reproduced data, the speakers are first clustered based on the x-vector characteristics [135], which are fixed dimensional embeddings that utterances with variable lengths are mapped to by a DNN. Then each audio is chunked into overlapping segments of 0.5 seconds each with a 50% overlap to minimize the distortions during reconstruction.

The RVAE is then trained on a given set, and from the new domain, speakers are then assigned to their appropriate clusters and new data is reconstructed via the previously trained RVAE. What is expected is the RVAE can only generate data similar to what it previously learned. So, it is expected that the data generated from a new different type of speech will be mapped to the same characteristics of the original training set.

9.3 Experimental Setup

Mel-Frequency Cepstrum Coefficients (MFCC) and the Mel Filterbank are two typical auditory aspects of ASR. Additionally, a bottleneck AE has been developed to extract a bottleneck feature. The availability of a RVAE-based feature vector is proven in this work in comparison to the availability of MFCC feature vectors in an ASR job. Rather than using voice waveforms as observation vectors, Fourier transform is used to extract the spectrum sequences from the speech waveforms.

A segmented waveform is created for a voice waveform sampled at 16 kHz and 16 bits at a certain time by utilizing a 512-point hamming window (frame shift, 10 ms). Then, using the Discrete Fourier Transform, the spectrum is computed from the waveforms (DFT). Real and imaginary numbers are treated separately in this research (i.e., a power spectrum is not used). As a result, a single RVAE unit (i.e., input vector) has 512 dimensions.

To begin, the encoder calculates the latent vector z from the input vector. The decoder then reconstructs the spectrum from the value of z. The spectrum is converted to the speech waveform using inverse DFT, and z is sampled using a Gaussian distribution. As a consequence of resampling, the values of z are altered. Then, by repeating the sampling procedure, we acquire slightly varied speech waveforms.

The Hispanic English data set was used for training the RVAE. The structure of the VAE used in this investigation and its training conditions are as follows: the number of layers is four for both encoder and decoder and have 512 nodes each. The number of nodes in this case is 512 (except for the output and the input layers of the encoder and decoder, respectively).

We analyzed latent vectors in four dimensions (2, 6, 13, and 20). The Hispanic English dataset was then enriched by a converted version of the TIMIT dataset with different portions ranging from 25% to 100%. The training conditions followed that, the acoustic models were trained using the Kaldi ASR toolset, and TIMIT was used in this experiment to train a GMM-HMM and a DNN-HMM for ASR. The following is the formula used to train the four different kinds of acoustic models.

1- Tri1: GMM-HMM estimate using maximum likelihood (ML)
2- Tri2: GMM-HMM feature transformation using Linear Discriminant Analysis (LDA) + Maximum Likelihood Linear Transform (MLLT).
3- Tri3: Feature-space-based GMM-HMM using Speaker Adaptive Training (SAT) Linear Regression with Maximum Likelihood (fMLLR) DNN.
4- DNN-HMM and further fine-tuning

Take note that the recipe includes a linguistic model (trigram with 70k words).

9.4 Results

The PERs obtained using the acoustic models trained with Hispanic-English speech and its augmented versions are represented in Table 9.1. As demonstrated, Data Augmentation has a marginal effect on the PER of the GMM-HMM model (tri1, tri2, and tri3) when compared to models trained using the Hispanic-English dataset alone. As observed in Table 9.1, the PERs of the reconstructed speech are a few percentage points higher than those of the original speech; consequently, DA through the RVAE had no effect on the refinement of the GMM-based models. On the other hand, the DNN model trained using the enhanced Hispanic dataset significantly improved the PER for (Hispanic-Eng+timit_reconstructed25) and (Hispanic-Eng+timit_reconstructed50). As a result, the suggested DA was effective in generating an accurate acoustic model. With TIMIT, the augmentation resulted in a relative improvement of 4.5 percent.

It is also noted that the RVAE data augmentation does not deteriorate the overall performance like the other proposed techniques. This can be due to the nature of the reconstruction itself, which is dependent on one domain of mapping rather than mixing more than one mapping function.

Table 9.1: TIMIT testing using reconstructed Hispanic-Eng. data from a TIMIT RVAE

Training Dataset (Percentage)	Model	PER
Hispanic-English	GMM-HMM-si	47.3
	GMM-HMM	40.1
	DNN-HMM	32.6
	LSTM-HMM	31.8
Hispanic-Eng. + Timit_reconstructed25%	GMM-HMM-si	46.3
	GMM-HMM	38.9
	DNN-HMM	27.1
	LSTM-HMM	25.3
Hispanic-Eng. + Timit_reconstructed50%	GMM-HMM-si	46.8
	GMM-HMM	39.3
	DNN-HMM	27.5
	LSTM-HMM	25.8
Hispanic-Eng. + Timit_reconstructed75%	GMM-HMM-si	46.6
	GMM-HMM	39
	DNN-HMM	27.1
	LSTM-HMM	25.1
Hispanic-Eng. + Timit_reconstructed100%	GMM-HMM-si	46.6
	GMM-HMM	39.4
	DNN-HMM	27
	LSTM-HMM	25.3

Again, here the LSTM-HMM show the best improvement when the data augmentation is used.

9.5 Conclusion and Future work

This section discussed how to supplement a speech corpus with RVAE-based data for acoustic modelling and examined the availability of a latent-based acoustic feature retrieved by the encoder in the RVAE for ASR. The experimental findings indicate that DA enhanced the performance of the ASR. Take note that this research used a RVAE. In the future, context data will be used for a multi-stage encoding, resulting in higher-quality synthesized speech. Additionally, a semi-supervised RVAE will be utilized to produce waveforms comprising a range of speakers' individualities.

Conclusion and Contributions

This thesis investigates the capacity of different ASR models to accommodate variabilities in speech. The first type of flexibility comes from the learning model itself. Yet, it is very difficult to control since incorporating more variabilities means requiring more modelling capacity which in turn needs more data. Of course, due to the expansion of ASR applications, along with the raised concerns on data privacy, the availability of more data becomes a challenge. Therefore, this thesis proposed three new data augmentation techniques that do not just add variations in properties, but in structure as well. The first of which is Neural Style Transfer, where the observations of one domain are mimicked by another domain producing interesting results. This type of augmentation is very effective, yet computationally demanding. The second is WaveNet generative modelling. It was used for TTS previously, but here it is investigated for data augmentation. This type needs to be trained only on the target speech, but again since itself needs data to be trained that can become an obstacle for using this method. This is best recommended when the new domain is close to the original. Last, the recurrent autoencoder, is somewhere in between both approaches in terms of computation requirements and quality of the results. Table 10.1 summarizes and compares the proposed techniques.

Table 10.1: Summary and comparison of proposed techniques for Data Augmentation

Proposed Technique	Model Used	Pros	Cons
Neural Style Transfer	Generative Adversarial	Produces high quality data	Requires a lot of data for training and matching speakers from the target domain
WaveNet Data Augmentation	Dilated Convolutional Network	Faster computations	Lesser quality, and requires more data from the target domain
Recurrent Variational Autoencoder	Variational Autoencoder	Requires no data from the target domain during training	Produces more stable results, but requires a lot of preprocessing

This Page Intentionally Left Blank

Synopsis

Revisiter les systèmes de reconnaissance automatique de la parole pour augmenter la capacité

11.1 Chapitre 1: Introduction

La popularité de la reconnaissance automatique de la parole (ASR) est sans aucun doute en plein essor, renforçant son importance dans la sphère technologique. La reconnaissance vocale a permis aux utilisateurs de se connecter avec ces appareils et gadgets de manière plus fluide, révolutionnant potentiellement l'environnement d'interaction homme-machine. Le but d'un système de reconnaissance vocale est de transformer correctement une forme d'onde audio (signal vocal) en mots via une interface informatique, quel que soit le locuteur ou les variables ambiantes. En d'autres termes, l'ASR est un système qui accepte un signal vocal en entrée et produit des mots qui correspondent au signal vocal d'entrée. Convertir un signal vocal en mots est une entreprise difficile car les signaux vocaux sont naturellement complexes. Il existe plusieurs variantes: physiologiques, environnementales, linguistiques, etc.

11.1.1 Défis

L'ASR présente de nombreux défis. Un signal de parole est simplement décomposé en source et filtre, la source étant les cordes vocales humaines dans la parole vocale et le filtre étant le conduit vocal et les articulateurs. La façon dont les gens ont tendance à prononcer les mots est appelée accents. Les accents font partie d'un dialecte qui englobe également la grammaire et le vocabulaire. Il est déjà assez difficile d'avoir un système ASR qui reconnaisse la parole avec différents accents, mais encore plus lorsque différents dialectes sont ajoutés. L'anglais étant la langue internationale du monde, il convient de mentionner comment les dialectes américains et britanniques peuvent différer distinctement dans les accents, grammaire et vocabulaire. Il existe environ 30 dialectes majeurs en Amérique, alors qu'il existe environ 40 dialectes au Royaume-Uni seulement. Malgré toutes les percées qu'ASR a réalisées au cours des quatre dernières décennies, il est toujours limité par de nombreuses limitations qui dégradent ses performances.

Les bruits, y compris la réverbération, peuvent constituer une contrainte importante sur les performances de l'ASR. Le bruit peut être classé comme stationnaire lorsqu'il est constant dans le temps, ou non

stationnaire lorsqu'il varie avec le temps, comme les événements sonores transitoires, les haut-parleurs interférents et la musique. Le bruit additif stationnaire à court terme peut être efficacement traité en utilisant des techniques de traitement de signal de réduction de bruit typiques et non supervisées tant qu'une détection fiable des instants du signal cible (parole) est réalisable. La réverbération est simplement la persistance ou le retard du son après la disparition de ce son; ce qui équivaut à une distorsion de la pièce. Détecter et atténuer les impacts des bruits ambiants non stationnaires, des sources sonores non stationnaires concurrentes ou des situations fortement réverbérâtes, en revanche, reste extrêmement difficile dans la réalité.

11.1.2 Organisation de la thèse

Cette thèse est organisée comme suit : ce chapitre donne un bref aperçu des progrès réalisés par les systèmes ASR, mais aussi des défis auxquels ils sont encore confrontés. Il donne également un bref aperçu de la synthèse pour l'augmentation des données où les données de formation sont enrichies de manière synthétique. Le chapitre 2 porte sur l'énoncé du problème, qui présente la portée de la thèse, ainsi que les objectifs de recherche, les questions de recherche et les hypothèses de recherche. Le chapitre 3 fournit des informations générales et des travaux connexes sur l'amélioration de l'ASR et l'augmentation des données. Le chapitre 4 traite des tentatives de la littérature dans l'amélioration de la parole. Le chapitre 5 traite des variations de la parole et présente différents ensembles de données utilisés dans cette étude, tandis que le chapitre 6 étudie les capacités du modèle dans les variations environnementales. Les chapitres 7, 8 et 9 discutent de trois techniques d'augmentation de données nouvellement proposées qui s'attaquent à la variabilité du locuteur. Les propositions sont Neural Style Transfer, WaveNet Modelling et Recurrent Autoencoders. Le chapitre 10 conclut ensuite les résultats de cette thèse. L'annexe A montre les défis auxquels nous avons été confrontés pendant la phase d'expérimentation, y compris de nombreuses expériences ratées. L'annexe B examine plus en détail la structure de la boîte à outils Kaldi et une idée générale de son fonctionnement.

11.2 Chapitre 2 : Énoncé du problème

La portée de cette thèse est de montrer comment les systèmes ASR de pointe couramment utilisés peuvent se faire; à savoir le modèle hybride de Markov caché - réseau de neurones profond (HMM-DNN), le modèle de mélange Gaussian de sous-espace (SGMM) et les réseaux de neurones récurrents de

séquence à séquence (de bout en bout) gèrent les variations dans un environnement de travail. Une approche d'enrichissement des jeux de données est également proposée en introduisant des variabilités synthétiques pour améliorer les performances des systèmes ASR lorsqu'ils travaillent dans un environnement réel.

11.2.1 Objectifs de recherche

Cette thèse s'intéresse à trouver des solutions aux variations environnementales entre les paramètres de formation et de travail des systèmes ASR. Un problème central avec les algorithmes de formation en apprentissage automatique est qu'ils ne généralisent pas bien les données sur lesquelles ils se sont entraînés. L'objectif principal est d'identifier le pouvoir de généralisation de différents systèmes ASR de pointe et leur capacité à apprendre à partir de petites portions de distorsions introduites dans l'ensemble d'apprentissage. De plus, une procédure d'augmentation de données qui introduit des variabilités communes synthétiques aux données est proposée pour améliorer les performances. Les objectifs de recherche détaillés sont :

1. Tester la robustesse de divers modèles et fonctionnalités de pointe par rapport aux variations entre les environnements de formation et de test.

2. Identifier la capacité de ces modèles et caractéristiques à apprendre à partir de données contenant de multiples sources d'interférences.

3. Identifier les améliorations de performance lors de l'introduction de plusieurs sources synthétiques de bruit dans l'environnement de formation.

11.3 Chapitre 3 : Contexte

11.3.1 Analyse de la parole et extraction de caractéristiques

La parole est générée via un système de voies vocales variant dans le temps, ce qui se traduit par des signaux de parole dynamiques ou variant dans le temps. Alors que le locuteur contrôle de nombreux composants de la production de la parole, y compris le volume, la voix, la fréquence fondamentale et la structure du conduit vocal, une grande partie de la variation de la parole est aléatoire, par exemple, la vibration des cordes vocales n'est pas entièrement périodique.

Le but de l'analyse de la parole est d'extraire des caractéristiques du signal de parole en le transformant en un autre signal ou un ensemble de signaux. Le modèle acoustique est utilisé pour approximer les aspects acoustiques de la parole; « Temporel » en agissant directement sur la forme d'onde de la parole ou « Spectral » dans le domaine fréquentiel après une transformation spectrale. Dans l'analyse dans le domaine temporel, l'accent est mis sur le fenêtrage court puisque la parole est hautement non stationnaire. Pendant le traitement de la parole, on suppose généralement que la parole est quasi-stationnaire (invariante dans le temps), donc un bon compromis serait une fenêtre d'analyse d'une longueur de 20 à 30 ms avec une mise à jour de 10 ms.

Lors de l'extraction de caractéristiques, nous nous efforçons d'extraire des caractéristiques résistantes au bruit à partir d'échantillons produits dans différents environnements sonores, car le bruit corrompt les échantillons et déforme les caractéristiques. C'est sur le front-end, comme pour le back-end, c'est un processus de modification des paramètres du modèle de parole ou de le changer complètement afin de traiter correctement les distorsions liées au bruit. Les échantillons de parole sont transformés en vecteurs de caractéristique de parole après avoir été échantillonnés au taux de Nyquist. Les caractéristiques extraites doivent être discriminantes pour discriminer les classes les unes des autres (suffisamment séparées dans l'espace) tout en étant robustes aux variations au sein d'une même classe. Après extraction, les caractéristiques sont dirigées vers le décodeur pour une reconnaissance correcte.

11.3.2 Modèles de Markov cachés (HMM)

Les signaux vocaux ont une structure temporelle et le travail du système ASR consiste à transformer des énoncés vocaux de longueur variable en séquences de mots de longueur variable. Puisque la parole est une séquence de sons dont les propriétés varient dans le temps, nous prenons de courts segments de parole (frames of speech) dont nous extrayons certaines caractéristiques qui nous aideront à reconnaître une identité sonore. La phase de traitement du signal pour extraire un ensemble de caractéristiques est appelée paramétrisation.

Les modèles de Markov cachés (HMM) fournissent un cadre statistique pour la modélisation acoustique du signal de parole. Les HMM convertissent les séquences d'observation (trames acoustiques) en séquences d'étiquettes (phonèmes). Les HMM offrent la distribution de probabilité sur toutes les séquences d'étiquettes potentielles pour une observation acoustique donnée. La parole est hautement non stationnaire et varie avec le temps ; cependant, les HMM modèlent la parole en supposant deux

circonstances; indépendance quasi-stationnaire et conditionnelle. Malgré le fait que les vecteurs de caractéristiques sont fortement corrélés, l'indépendance conditionnelle suppose que les vecteurs de caractéristiques sont conditionnellement indépendants de ceux qui sont avant et après, la probabilité de transition d'état suivant dépendant uniquement de l'état actuel.

11.3.3 Modèles de mélange Gaussian (GMM)

Les GMM sont une notion probabiliste utilisée pour catégoriser les données en fonction de leur distribution de probabilité. Le GMM peut être utilisé pour représenter n'importe quel ensemble de données utilisant plusieurs distributions Gaussiannes. De plus, ils peuvent être utilisés pour identifier des clusters dans des ensembles de données lorsque les clusters ne sont pas bien définis. Le modèle de mélange Gaussian est un modèle probabiliste dans lequel tous les points de données sont supposés être créés à l'aide d'un mélange de distributions Gaussiannes. De plus, les GMM peuvent être utilisés pour déterminer la probabilité qu'un nouveau point de données soit membre de chaque cluster. Les GMM sont relativement immunisés contre les valeurs aberrantes, ce qui signifie qu'ils peuvent toujours fournir des résultats précis même si certains points de données ne correspondent pas parfaitement à l'un des clusters. En conséquence, les GMM sont un outil polyvalent et puissant pour le regroupement de données. Il peut être considéré comme un modèle probabiliste dans lequel chaque groupe est censé avoir des distributions Gaussiannes avec des moyennes et des covariances qui déterminent ses paramètres. Les GMM sont composés de deux composants : des vecteurs moyens et des matrices de covariance. Une distribution de probabilité Gaussianne est une distribution continue qui prend la forme d'une courbe en cloche.

Les GMM sont souvent utilisés pour décrire la distribution de probabilité de mesures ou de caractéristiques continues dans un système biométrique, telles que les paramètres spectraux liés aux voies vocales dans un système d'identification du locuteur. Les paramètres GMM sont estimés à partir de données d'entraînement à l'aide de l'approche itérative d'espérance-maximisation (EM) ou à partir d'un modèle antérieur bien entraîné utilisant l'estimation Maximum A Posteriori (MAP).

11.3.4 Réseaux de neurones profonds (DNN)

Les DNN sont une autre alternative de distribution de sortie d'état avec la capacité de modéliser des relations non linéaires complexes. Les DNN ont la capacité de dériver des représentations internes discriminatoires à partir de signaux vocaux qui résistent aux différentes sources de variabilité. Avec

l'augmentation de la profondeur du réseau, ces représentations deviennent moins sensibles aux petites perturbations d'entrée, ce qui améliore les performances de reconnaissance vocale. Les DNN sont généralement des réseaux de neurones à anticipation, ils peuvent cependant être entraînés par rétropropagation, ce qui nous permet de calculer et d'attribuer les erreurs associées à chaque neurone. Cela nous permet de modifier et d'adapter efficacement les paramètres du modèle.

Pour évaluer la précision du modèle lors de son apprentissage, une fonction de coût est utilisée. La fonction de coût doit être minimisée afin de garantir que le modèle s'adapte correctement à chaque observation. Tout en ajustant les poids et le biais du modèle, il utilise la fonction de coût et l'apprentissage par renforcement pour atteindre le point de convergence, ou minimum local. La méthode modifie ses poids par descente de gradient, ce qui permet au modèle de trouver la meilleure direction à prendre pour minimiser les erreurs ; minimiser la fonction de coût. Les paramètres du modèle sont ajustés à chaque cas d'apprentissage afin de converger progressivement vers le minimum. Les DNN doivent prendre en compte divers facteurs de formation, notamment le nombre de couches, les unités par couche, le taux d'apprentissage et les poids initiaux. Avec un nombre élevé de paramètres, les réseaux de neurones profonds sont des systèmes d'apprentissage automatique très puissants. Cependant, le surajustement est un problème important avec ces réseaux. Le décrochage est une stratégie pour résoudre ce problème. Pendant la formation, le concept de base consiste à supprimer au hasard des unités, en les laissant tomber avec leurs connexions du réseau de neurones. Cela élimine efficacement le surajustement.

11.3.5 Hybride DNN-HMM

Sans aucun doute, les modèles de Markov cachés de réseau de neurones profonds (DNN-HMM) ont montré qu'ils peuvent surpasser les modèles de Markov cachés basés sur un modèle de mélange Gaussian en termes de reconnaissance vocale (GMM-HMM). Le système hybride DNN-HMM, tel qu'illustré à la Figure 11.1, exploite la capacité d'apprentissage de représentation de DNN et la capacité de modélisation séquentielle de HMM sur une variété de tâches de reconnaissance audio continue à grand vocabulaire. L'hybridation des DNN et des HMM a eu des résultats prometteurs dans la reconnaissance de la parole et de l'image, contrairement aux premiers stades où les réseaux de neurones remplaçant les GMM consistaient en une seule couche cachée, ce qui rendait difficile l'adaptation et prenait trop de temps à s'entraîner. Les observations de plusieurs trames consécutives sont introduites en tant qu'entités d'entrée dans une séquence de couches cachées. Les HMM sont utilisés pour représenter la dynamique du signal de parole, tandis que les DNN sont utilisés pour estimer les probabilités d'observation. Chaque neurone

de sortie DNN est formé pour prédire la probabilité a posteriori de la densité continue de l'état des HMM compte tenu des données acoustiques. Le modèle DNN convertit les caractéristiques de la parole en étiquettes correspondant aux états de Markov cachés dans les modèles de Markov cachés (HMM). Outre sa discrimination intrinsèque, les DNN-HMM offrent deux avantages supplémentaires : l'apprentissage peut être effectué à l'aide de l'algorithme de Viterbi intégré, et le décodage est généralement très efficace.

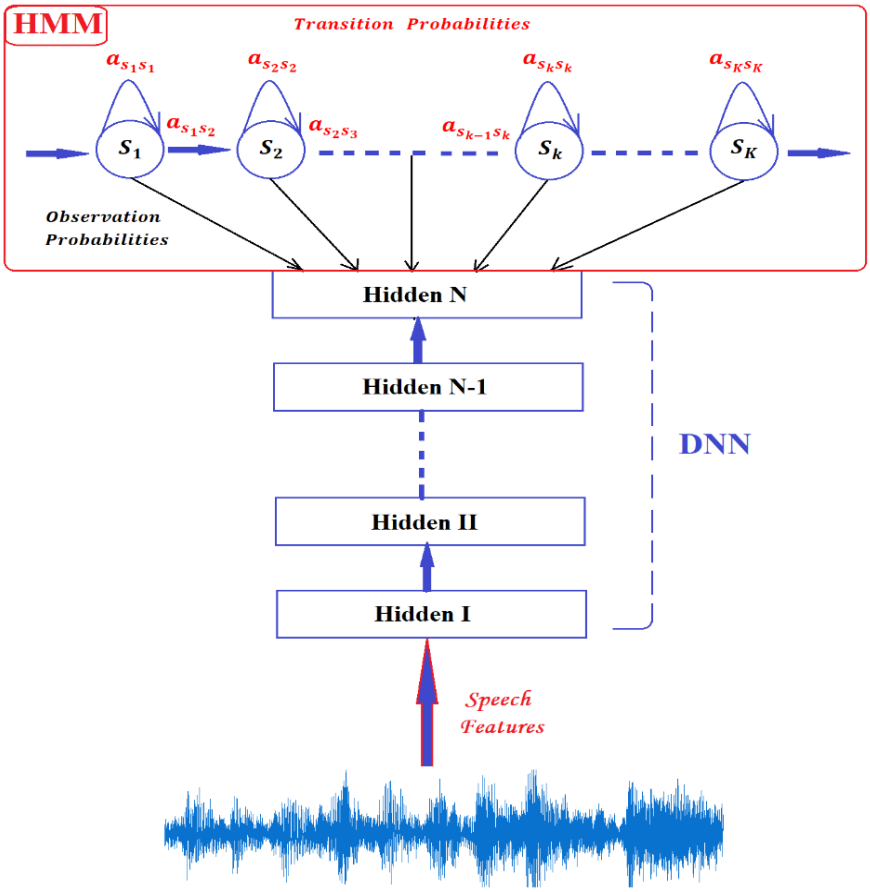


Figure 11.1: Structure DNN-HMM

11.3.6 Réseaux de neurones convolutifs (CNN)

Les CNN sont largement utilisés en intelligence artificielle en raison des avancées significatives qu'ils ont réalisées, notamment dans le traitement des ensembles de données séquentielles. La popularité des CNN découle de sa capacité à extraire automatiquement une hiérarchie de fonctionnalités robustes, améliorant ainsi les performances. La structure d'un CNN neuronal conventionnel est composée d'une ou plusieurs couches convolutives, d'une couche de regroupement maximum et d'une couche entièrement connectée

suivie d'une rétropropagation. La structure est composée de nombreux neurones, avec plusieurs clones identiques du même neurone dans chaque couche convolutive, ce qui offre aux CNN l'avantage de ne pas nécessiter un grand nombre de paramètres réels pour les calculs de grands modèles. Le principal avantage des CNN est qu'ils réduisent considérablement le nombre de paramètres dans les ANN. L'inconvénient de l'ANN est le grand nombre de paramètres de poids qui doivent être entraînés.

Le concept de base des réseaux de neurones convolutifs est l'utilisation de filtres. Un filtre est une matrice qui agit comme une fenêtre coulissante à l'intérieur d'une image et est responsable de la reconnaissance des caractéristiques ou des motifs dans les images à l'aide de la convolution. Plutôt que d'examiner l'intégralité de l'image, les filtres recherchent des sections spécifiques. Une image a des bords, des formes et des couleurs, et quand tout cela est combiné, elle a des caractéristiques. Les filtres convoluent l'entrée et transmettent la sortie résultante à une couche de regroupement. Encore une fois, la sortie de la couche de regroupement peut être acheminée via un mélange de couches de convolution et de regroupement. Enfin, il est transmis à une couche entièrement connectée.

11.3.7 Réseaux de neurones convolutifs unidimensionnels (CNN 1D)

Comme indiqué précédemment, les CNN sont optimisés pour les données bidimensionnelles. Un CNN 1D est un modèle plus approprié pour les données 1D, telles que les signaux audio. Dans les CNN 1D, le noyau se déplace dans une seule direction, ce qui entraîne des données d'entrée et de sortie unidimensionnelles, tandis que dans les CNN 2D, le noyau se déplace dans deux directions. Pour les données de série 1D telles que les signaux audio, un CNN 1D surpasse un CNN 2D normal. Les données d'entrée unidimensionnelles sont transmises à un mélange de couches de convolution et de regroupement unidimensionnelles. La sortie de la dernière couche de regroupement/convolution est ensuite envoyée à des couches entièrement connectées, qui fournissent une sortie finale prête pour la classification.

Non seulement un CNN 1D a besoin de moins de paramètres pour s'entraîner, mais le coût de calcul est bien inférieur à celui d'un CNN 2D [34]. Lorsqu'une image de taille *NxN* est convoluée avec un noyau de taille KxK à l'aide d'un réseau de neurones convolutionnel bidimensionnel, la complexité de calcul approche $O(N^2K^2)$. D'autre part, le coût de calcul du traitement des données de séries à N dimensions et de la convolution avec un noyau à K dimensions est d'environ $O(NK)$, ce qui est beaucoup moins. De plus, pour chaque taille de noyau, le nombre de paramètres à former est réduit à K à partir de K^2. En conséquence, un CNN 1D est bien adapté pour traiter les séries temporelles représentées par des signaux de données audio.

11.3.8 WaveNet

WaveNet est un concept relativement nouveau avec un grand potentiel. Il a la capacité de produire un discours authentique imitant la voix d'un humain. WaveNet est un réseau neuronal convolutif profond utilisé pour générer des formes d'onde audio brutes. WaveNet est capable de générer une parole qui semble plus authentique et imite avec précision n'importe quelle voix humaine que les meilleurs systèmes de synthèse vocale existants. WaveNet modélise la forme d'onde brute d'un flux audio, un échantillon à la fois. Il peut reproduire n'importe quel type d'audio, ce qui donne un discours plus naturel. Un seul WaveNet peut enregistrer avec précision les caractéristiques d'un large éventail de locuteurs et faire la transition entre eux en fonction de l'identification du locuteur.

Pour faire face aux dépendances temporelles à longue portée nécessaires à la création audio brute, WaveNet est structuré autour de convolutions causales dilatées avec des champs récepteurs très larges. La convolution dilatée peut être considérée comme une convolution avec des filtres appliqués sur une zone plus grande que sa longueur en omettant certaines valeurs d'entrée. Il permet au réseau de fonctionner à une échelle rudimentaire. Cela peut être comparé à la mise en commun ou à la foulée. La logique est que le résultat d'un seul pas de temps peut dépendre d'une séquence d'entrées plus longue d'entrées de pas de temps précédentes.

11.3.9 Mémoire longue à court terme (LSTM)

Un réseau de mémoire longue à court terme (LSTM) est un réseau de neurones récurrent (RNN), qui sont essentiellement des réseaux avec des boucles. Les RNN ont connu un grand succès dans la reconnaissance vocale, la modélisation du langage, la traduction, etc. Malheureusement, les RNN n'apprennent pas bien à connecter les informations lorsqu'il existe de très longs décalages, car ils souffrent d'une descente de gradient qui disparaît pendant la rétropropagation, ainsi qualifiée de mémoire courte. C'est là qu'interviennent les réseaux LSTM. Il s'agit d'un type spécial de RNN capable d'apprendre les dépendances à long terme. Les LSTM, comme leur nom l'indique, sont capables de conserver à la fois la mémoire courte et la mémoire longue.

La capacité de mémoriser des informations pendant de longues périodes est un comportement par défaut des réseaux LSTM. Un réseau LSTM est capable de classer, de traiter et de prédire des séries chronologiques lorsqu'il existe de très longs délais de taille inconnue entre les événements, surpassant les

RNN et les HMM. En ce qui concerne la reconnaissance automatique de la parole (ASR), les LSTM ont atteint un taux d'erreur de phonème minimal de 17,7 % sur l'ensemble de données de parole naturelle TIMIT classique en 2013. Un model répétitif dans LSTM contient quatre couches en interaction. La Figure 11.2 illustre la conception de l'unité LSTM. Nous remarquons la ligne horizontale "état de la cellule", qui est la clé de LSTM. Cette ligne d'état cellulaire traverse toute la chaîne avec des interactions linéaires mineures. Les LSTM peuvent effectuer des défis sophistiqués et artificiels à long décalage que les algorithmes de réseau récurrents antérieurs n'ont jamais été en mesure d'accomplir.

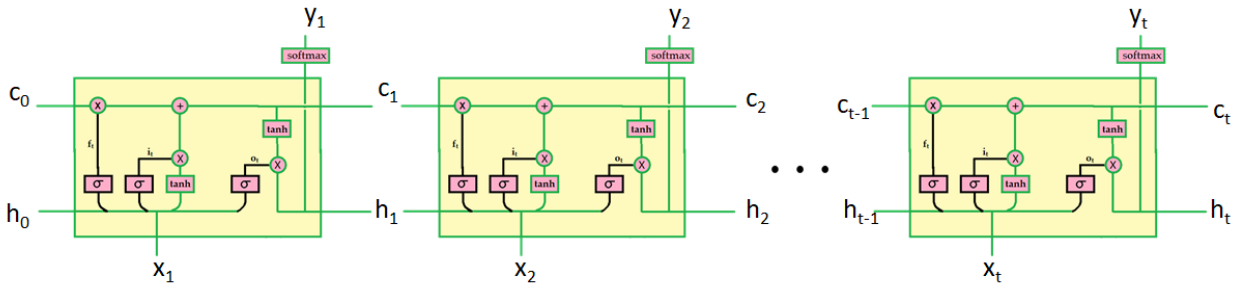


Figure 11.2: Architecture d'unité LSTM

11.3.10 Hyper-paramètres dans les algorithmes d'apprentissage automatique

Les modèles d'apprentissage automatique ont des paramètres qui peuvent être entraînés et personnalisables. Les paramètres pouvant être entraînés sont ceux dont les valeurs sont modifiées automatiquement pendant le processus de formation du modèle, tandis que les paramètres réglables sont réglés ou ajustés manuellement ou via l'utilisation d'une approche de réglage automatique avant la formation du modèle. Les paramètres pouvant être entraînés incluent les pondérations et le biais d'un réseau de neurones. D'autre part, les paramètres configurables ou réglables incluent les taux d'apprentissage, les époques, etc. Dans l'apprentissage automatique, ces paramètres personnalisables sont appelés hyperparamètres.

A cet égard, les hyperparamètres sont considérés comme externes au modèle puisque leurs valeurs ne peuvent pas changer pendant l'apprentissage ou pendant la formation. Ils sont établis avant le début de la phase de formation. Bien qu'ils soient utilisés pour former le modèle d'apprentissage automatique, ils ne sont pas inclus dans le modèle résultant. Les paramètres entraînables, d'autre part, pourraient être appelés un modèle résultant.

La configuration hyperparamétrique d'un algorithme affecte dans de nombreux cas ses performances sur un travail d'apprentissage spécifique. Les experts en apprentissage automatique peuvent affiner les hyperparamètres pour optimiser les résultats. Lavesson et Davidsson affirment que l'ajustement des hyperparamètres est souvent plus crucial que la technique d'apprentissage automatique utilisée. Les hyperparamètres ont un impact significatif sur les performances de tout modèle d'apprentissage automatique sur un ensemble de données donné. Une optimisation appropriée des hyperparamètres peut entraîner une amélioration importante des performances.

11.3.11 Boîte à outils Kaldi

Kaldi est une boîte à outils gratuite et open source pour les chercheurs en reconnaissance vocale. La boîte à outils a été créée pour la première fois en 2009 à l'Université John Hopkins par Daniel Povey et ses collègues. Il est développé en C++ et est extensible. Kaldi fournit un support important pour l'algèbre linéaire et peut générer des fonctionnalités telles que mfcc, fbank et fMLLR. Ainsi, dans la recherche actuelle sur les réseaux neuronaux profonds, Kaldi est souvent utilisé pour pré-traiter les formes d'onde brutes en caractéristiques acoustiques pour les modèles neuronaux de bout en bout.

11.4 Chapitre 4 : Tentatives de littérature dans l'amélioration de la parole

11.4.1 Transformateur augmenté par convolution pour la reconnaissance vocale

Combiner les CNN avec l'architecture Transformer basée sur l'auto-attention pour modéliser les dépendances locales et globales résulte en un bloc Conformer distinctif (transformateur augmenté par convolution). Le bloc Conformer comprend une pile d'un model d'anticipation, un modèle d'auto-attention, un modèle de convolution et se termine par un deuxième model d'anticipation. Le modèle de convolution empilé après le modèle d'auto-attention est le plus adapté à la reconnaissance vocale. D'autres méthodes de recherche adoptent le model de compression et d'excitation pour capturer un contexte plus long, mais sont encore limitées dans la capture du contexte global.

Conformer obtient des résultats de pointe impressionnants sur "LibriSpeech" avec une amélioration de 15 % par rapport au meilleur Transformer Transducer publié sur l'ensemble de données test-other avec un modèle de langage externe. Le Conformer atteint le WER le plus bas avec l'ajout d'un modèle de

langage. Le conformateur a montré des résultats intéressants par rapport à d'autres modèles. Le Table 11.1 compare le résultat du conformateur avec d'autres modes.

Table 11.1: Modèle conformateur sur LibriSpeech

Method	Params (M)	WER without LM		WER with LM	
		Test clean	Test other	Test clean	Test other
Hybrid					
Transformer [51]	-	-	-	2.26	4.85
LAS					
Transformer [52]	270	2.89	6.98	2.33	5.17
Transformer [53]	-	2.2	5.6	2.6	5.7
LSTM	360	2.6	6	2.2	5.2
Transducer					
Transformer [54]	139	2.4	5.6	2	4.6
ContextNet (S) [55]	10.8	2.9	7	2.3	5.5
ContextNet (M) [55]	31.4	2.4	5.4	**2**	4.5
ContextNet (L) [55]	112.7	**2.1**	4.6	**1.9**	4.1
Conformer					
Conformer (S)	10.3	**2.7**	**6.3**	**2.1**	**5**
Conformer (M)	30.7	**2.3**	**5**	**2**	**4.3**
Conformer (L)	118.8	**2.1**	**4.3**	**1.9**	**3.9**

11.4.2 SpecAugment : méthode d'augmentation des données pour l'ASR

Cette méthode d'augmentation de données est appliquée directement aux entrées de caractéristiques d'un réseau de neurones comme les coefficients du banc de filtres. Dans cette méthode, SpecAugment est appliqué de bout en bout sur les réseaux Listen, Attend et Spell (LAS), surpassant tous les travaux précédents sur LibriSpeech 960h et Switchboard 300h. Le Table 11.2 et le Table 11.3 affichent les résultats sur Librispeech 960h et Switchboard 300h respectivement par rapport aux résultats les plus proches des autres modèles. SpecAugment fonctionne sur le spectrogramme log-mel de l'audio d'entrée et non sur l'audio brut, comme s'il s'agissait d'une image. Il se compose de trois déformations; Déformation temporelle, masquage temporel et masquage fréquentiel. Malgré la simplicité de cette méthode, elle est très efficace et peu coûteuse en temps de calcul.

Il a montré des performances de pointe même sans modèle de langage (LM). Le programme de taux d'apprentissage ainsi que l'augmentation peuvent maximiser les performances du réseau. Dans cette

méthode d'augmentation, des horaires longs et très longs sont introduits, et un modèle de langage RNN par fusion superficielle est adopté. Les résultats surpassent les systèmes hybrides précédents. Des performances de pointe sont obtenues même sans modèle de langage. De plus, l'entraînement avec augmentation présente un avantage majeur; les réseaux sous-ajustent la perte et le WER, convertissant un problème de sur-ajustement en un problème de sous-ajustement. Lors de la résolution du sous-ajustement par des approches standard, des gains de performance significatifs ont été obtenus.

Table 11.2: SpecAugment sur LibriSpeech 960h WER (%)

LibriSpeech 960h WERs (%)				
Method	**No LM**		**With LM**	
	clean	other	clean	other
HMM				
Yang et al., 2018	-	-	2.97	7.5
CTC/ASG				
Li et al., 2019 [60]	3.6	11.95	2.95	8.79
LAS				
Irie et al., 2019 [61]	4.7	13.4	3.6	10.3
Sabour et al., 2019	4.5	13.3	-	-
SpecAugment				
LAS	4.1	12.5	3.2	9.8
LAS + SpecAugment	**2.8**	**6.8**	**2.5**	**5.8**

Table 11.3: SpecAugment sur Switchboard 300h WERs(%)

Switchboard 300h WERs (%)				
Method	**No LM**		**With LM**	
	clean	other	clean	other
HMM				
Zeyer et al., 2018	-	-	8.3	17.3
LAS				
Weng et al., 2018 [62]	12.2	23.3	-	-
Zeyer et al., 2018 [63]	11.9	23.7	11	23.1
SpecAugment				
LAS	11.2	21.6	10.9	19.4
LAS + SpecAugment (SM)	**7.2**	14.6	**6.8**	14.1
LAS + SpecAugment (SS)	7.3	**14.4**	7.1	**14**

11.4.3 Formation d'étudiants bruyante (NST)

L'idée est d'adopter pour l'ASR une méthode d'apprentissage semi-supervisée existante qui a montré des améliorations de classification d'images connue sous le nom de "Noisy Student Training". Le principe original de cette méthode est l'auto-formation itérative qui tire parti de l'augmentation. NST utilise des données non étiquetées pour améliorer la précision. Pour adapter cette méthode à l'ASR, les éléments suivants ont dû être utilisés :

- Adaptive SpecAugment est présenté comme la méthode d'augmentation qui agit sur le spectrogramme de l'entrée audio, en utilisant le masquage temporel adaptatif.
- Fusion de modèles linguistiques sur le réseau d'enseignants, générant de meilleurs relevés de notes pour le réseau d'étudiants à former. Un score de filtrage normalisé est proposé pour les transcriptions.
- Échantillonnage sous-modulaire pour équilibrer le jeu de données en pondérant les échantillons générés par le réseau d'enseignants (couples énoncé-relevé de notes).
- Un score de filtrage normalisé pour les relevés de notes créés par les réseaux d'enseignants qui est proportionnel au score de fusion et au nombre de jetons.

La première série d'expériences a été menée sur LibriSpeech 100-860, où le sous-ensemble propre de 100h est utilisé comme données étiquetées et les 860h restantes comme données non étiquetées. Les résultats sont représentés dans le Table 11.4**.** La deuxième série d'expériences a été menée sur LibriSpeech-LibriLight, dans laquelle le Librispeech complet est utilisé comme données étiquetées et le sous-ensemble unlab-60k comme données non étiquetées.

Le Table 11.5 affiche les résultats. Il a été conclu que l'utilisation de la filtration graduelle pour les performances à faible supervision LibriSpeech 100-860 était bénéfique, alors que peu d'avantages pour les performances à haute supervision LibriSpeech-LibriLight

Table 11.4: Formation des étudiants bruyants sur LibriSpeech 100h WERs (%)

LibriSpeech 100h WERs (%)				
Method	**Dev**		**Test**	
	clean	other	clean	other
Supervised				
Luscher et al., 2019 [67]	5	19.5	5.8	18.6
Semi-supervised (w/ LibirSpeech 860h)				
Hsu et al., 2019 [68]	5.39	14.89	5.78	16.27
Ling et al., 2019 [69]	-	-	4.74	12.2
Proposed Method				
Baseline (LAS + SpecAugment)	5.3	16.5	5.5	16.9
+ NST before LM Fusion	4.3	9.7	4.5	9.5
+ NST with LM Fusion	**3.9**	**8.8**	**4.2**	**8.6**

Table 11.5: Formation d'étudiants bruyants sur LibriSpeech 960h WER (%)

LibriSpeech 960h WERs (%)				
Method	**Dev**		**Test**	
	clean	other	clean	other
Supervised				
Zhang et al., 2020 [54]	-	-	2	4.6
Han et al., 2020 [55]	1.9	3.9	1.9	4.1
Semi-supervised				
Synnaeve et al., 2018 [52]	2	3.65	2.09	4.11
Proposed Method with baseline				
ContextNet + NST before LM Fusion	**1.6**	3.7	**1.7**	3.7
ContextNet + NST after LM Fusion	**1.6**	**3.4**	**1.7**	**3.4**

11.4.4 Augmentation des données "Text-To-Speech"

La méthode d'augmentation est une technique d'apprentissage semi-supervisée. Pour un sous-ensemble de données de formation de 100 heures, les systèmes TTS et ASR de base sont formés séparément. Ensuite, en utilisant un sous-ensemble plus large, les énoncés sont synthétisés et utilisés comme données d'apprentissage ASR. Le transformateur de la boîte à outils de reconnaissance vocale ESPNET est utilisé pour le modèle ASR.

Le transformateur est une architecture séquence à séquence composée de deux réseaux de neurones (encodeur et décodeur). Le modèle de langage (LM) utilisé est un réseau neuronal récurrent (RNN) composé de quatre couches LSTM. Les énoncés trop longs ou trop courts sont supprimés et les données

sont perturbées par la vitesse, ce qui rend l'ensemble d'apprentissage 3 fois plus grand. Tacotron est sélectionné comme synthétiseur vocal de base, ce qui crée un spectrogramme Mel à 80 bandes de magnitude logarithmique à partir du test d'entrée. Les caractéristiques acoustiques sont augmentées avec SpecAugment pendant la phase de formation.

Le système TTS est une configuration à deux réseaux ; un synthétiseur (texte d'entrée vers spectrogramme) et un vocodeur (spectrogramme vers forme d'onde). Lorsque la parole synthétisée est ajoutée, la ligne de base ASR de bout en bout a obtenu une amélioration de 39 % de WER relatif sur le test-clean et de 21 % sur le test-other. Le Table 11.6 montre que la méthode proposée surpasse uniquement l'ensemble de formation clean-460 et surpasse les autres dans la configuration des ressources faibles à moyennes, comme indiqué dans le Table 11.7.

Table 11.6: TTS Augmentation on LibriSpeech train-clean-100/460 [70]

Training Set	***ASR System***	***WER (%)***			
		dev		**test**	
		clean	other	clean	other
clean-100	Kaldi	5.9	20.4	6.6	22.5
	RETURNN Hybrids	**5**	**19.5**	**5.8**	**18.6**
	E2E proposed method	10.3	24	11.2	24.9
clean-460	Kaldi	5.3	17.7	5.8	19.1
	E2E proposed method	**4.5**	**14.1**	**5.1**	**14.1**

Table 11.7: TTS Augmentation in comparison with other works [70]

Setup	***Other works***	***WER (%)***		***WER Improvement(%)***	
		dev		**test**	
		clean	other	clean	other
Low-to-Medium Resource	**Proposed method**	**4.3**	**13.5**	**38.6**	**20.6**
	Method [73]	9.3	30.6	22.8	10.1
	Method [74]	5.4	22.2	33.3	9.4
Medium Resource	**Proposed method**	**3.2**	**9.1**	**8.6**	0
	Method [73]	6.3	22.5	0.3	-0.5
Large Resource	Method [75]	4.7	15.5	**8.6**	**4.6**
	Method [73]	4.6	13.6	4.6	1.8
	Method [74]	**2.5**	**7.2**	4.9	2.4

11.4.5 Augmentation des données "On The Fly"

Les modèles séquence à séquence (S2S) sont capables de performances de pointe si le surajustement est évité. Pour ce faire, les données de formation doivent être augmentées par une augmentation de données lorsque les ressources de données sont faibles, comme:

- Étirement dynamique du temps : la séquence d'entrée est modifiée en manipulant la série chronologique des vecteurs de fréquence (caractéristiques) pour obtenir l'effet de perturbation de la vitesse. Chaque fenêtre de vecteur de caractéristiques est étirée à l'aide de l'interpolation du plus proche voisin.
- SpecAugment : L'entrée du spectrogramme est modifiée avec un masquage de fréquence et de temps avant d'être transmise au modèle S2S.
- Échantillonnage de sous-séquences (échantillonnage sous contrainte) : étant donné un énoncé, trois variantes différentes de sous-séquences sont autorisées ; avec une répartition égale.

Deux modèles de séquence à séquence différents sont utilisés ; S2S basé sur LSTM et S2S d'auto-attention. Les deux modèles peuvent être améliorés en combinant deux stratégies d'augmentation dans un seul entraînement (par exemple, en utilisant d'abord Time Stretching, puis SpecAugment pour les séquences d'entrée). Cela suggère que les deux stratégies aident à la généralisation des modèles à travers divers aspects et peuvent être utilisées conjointement l'une avec l'autre.

Comme représenté dans le Table 11.8 et le Table 11.9, la combinaison de modèles basés sur LSTM et d'auto-attention s'est avérée assez efficace pour réduire le WER. Lorsqu'aucun modèle de langue supplémentaire n'est utilisé, des performances de pointe sont atteintes sur les ensembles de test Switchboard et CallHome (CH).

Table 11.8: Augmentation "On The Fly" sur 2000h SWB + Fisher

2000h Switchboard + Fisher			
Model	**LM**	**SWB**	**CH**
Povey et al., 2016	n-gram	8.5	15.3
Saon et al., 2017	LSTM	5.5	10.3
Han et al., 2018	LSTM	5	9.1
Weng et al., 2018	-	8.3	15.5
Audhkhasi et al., 2018	-	8.8	13.9
LSTM-based (no Augmentation)	-	7.2	13.9
Transformer (no Augmentation)	-	7.3	13.5
LSTM-based	-	5.5	11.4
Transformer	-	6.2	11.9
ensemble	-	5.2	10.2

11.4.6 Augmentation du locuteur à faible ressource

L'idée ici est d'utiliser une technique d'augmentation du locuteur pour synthétiser les données avec suffisamment de diversité de locuteurs et de textes. Un schéma d'augmentation du locuteur est proposé qui entraîne un modèle de synthèse vocale de bout en bout Tacotron2, conditionné cependant aux représentations du locuteur à partir d'un auto-encodeur variationnel (VAE). De plus, un classificateur de locuteur qui utilise des variables latentes comme entrée est formé conjointement. En conséquence, l'encodeur audio est plus susceptible de produire des variables latentes contenant des informations sur le locuteur, ce qui facilite la convergence du modèle.

Le modèle TTS peut synthétiser la parole de nouveaux locuteurs inconnus à l'aide de ces méthodes, en échantillonnant à partir de la distribution latente enseignée et en fournissant suffisamment de variations de locuteur dans la parole synthétique pour l'augmentation des données. Plus il y a de locuteurs virtuels échantillonnés, meilleures sont les performances ASR. Le Table 11.9 montre les résultats sur le standard 50h.

Table 11.9: Augmentation du locuteur à faible ressource sur 50 h SWB

Switchboard					
Model	**Data**	**Aug**	**Virtual Spkr**	**SWBD**	**CH**
	50 h	None	-	25.6	39.2
	50 h	SpecAugment	-	20.2	32.5
	50 h	TTS	0	21.1	35.2
Baseline	50 h	TTS + SpecAugment	0	17.8	29.7
	50 h	TTS + SpecAugment	25	17.2	28.8
Proposed	50 h	TTS + SpecAugment	100	16.5	28.7
	50 h	TTS + SpecAugment	300	**16.5**	**28.2**

11.5 Chapitre 5 : Variations de la parole et ensembles de données

11.5.1 Côté haut-parleur

- Les différents dialogues ou accents des différents locuteurs.
- L'état émotionnel de locuteur.
- L'état physique du locuteur.

11.5.2 Canal de communication

- Le milieu environnant.
- Existence de réverbération
- Existence de bruit non structuré

11.5.3 Jeux de données utilisés

- Bramshill
- Hispanic-English
- Nationwide Speech Project (NSP)
- TIMIT
- NTIMIT

11.6 Chapitre 6 : Effets sur la capacité du modèle dans la modélisation de données synthétiques pour la variabilité environnementale

Dans ce chapitre, nous examinons la capacité de différents modèles acoustiques ASR à maintenir les performances lorsqu'ils sont entraînés avec des données TIMIT et testés à l'aide d'une version déformée du réseau de TIMIT. La prise en compte d'un certain type de variabilité se fait en modifiant ou en adaptant l'un des composants du système ASR, atténuant ainsi l'effet de la variabilité dans des scénarios réels. Le tableau 6.1 répertorie les différents composants représentant certaines variabilités au sein d'un système ASR et les exemples correspondants de solutions dans la littérature

Robustesse au bruit dans différents composants ASR

Composant ASR	Utilisation exemplaire pour la robustesse
Données d'entraînement	Synthèse vocale pour l'augmentation des données
Pré-traitement	Filtrage et modélisation du bruit
Extraction de caractéristiques	Caractéristiques robustes au bruit
Modélisation acoustique/langage	Adaptation au locuteur et adaptation au domaine linguistique
Post-traitement	Post-traitement du langage et fusion de classificateurs

La capacité de tout modèle d'apprentissage automatique est proportionnelle à sa complexité. Néanmoins, une complexité croissante signifie généralement un nombre accru de paramètres apprenables et d'hyper-paramètres, qui sont des paramètres qui doivent être sélectionnés par le concepteur du modèle et qui ne sont pas appris à partir des données, nécessitant beaucoup de données pour apprendre correctement les données. Dans cette étude, nous allons examiner quatre vertus de différents modèles acoustiques ASR concernés par l'écart entre les environnements d'entraînement et de test :

- Généralisation lorsque l'environnement de test est significativement différent de l'environnement de formation.
- Adaptation pour apprendre à partir de données exemplaires fractionnaires.
- Performances spécialisées lorsqu'il n'y a pas de divergence entre les environnements de formation et de test.
- Robustesse face aux différents niveaux de bruit dans l'environnement de test.

Les ensembles de données utilisés pour les expériences sont le TIMIT, le NTIMIT et le TIMIT déformé par le bruit réel en utilisant l'ensemble de données QUT-NOISE. Pour évaluer les performances de la modélisation acoustique uniquement, les performances sont mesurées par le taux d'erreur téléphonique (PER) et non par le taux d'erreur de mots (WER) pour supprimer l'effet du modèle de langue sur les performances du système. Les modèles qui vont être étudiés sont le modèle de base de mélange Gaussian (GMM)-HMM, les modèles améliorés utilisant la transformation linéaire de vraisemblance maximale (MLLT) et l'analyse discriminante linéaire (LDA), l'hybride HMM - Deep Neural Network (DNN-HMM), et Subspace Gaussian Mixture Model (SGMM). La formation du modèle est effectuée à l'aide de la boîte à outils KALDI.

La Figure 11.3 montre graphiquement le taux d'amélioration des différents modèles par rapport à la modification du pourcentage de similarité des données d'entraînement avec les données de test. On peut directement noter qu'à mesure que la part des données induites du NTIMIT dans les données d'apprentissage TIMIT augmente, les performances de tous les modèles s'améliorent, mais à des rythmes différents. Tous les modèles connaissent une amélioration rapide lorsqu'ils sont introduits avec la première partie de l'ensemble de données à partir des données NTIMIT, puis arrivent à saturation après la sixième partie.

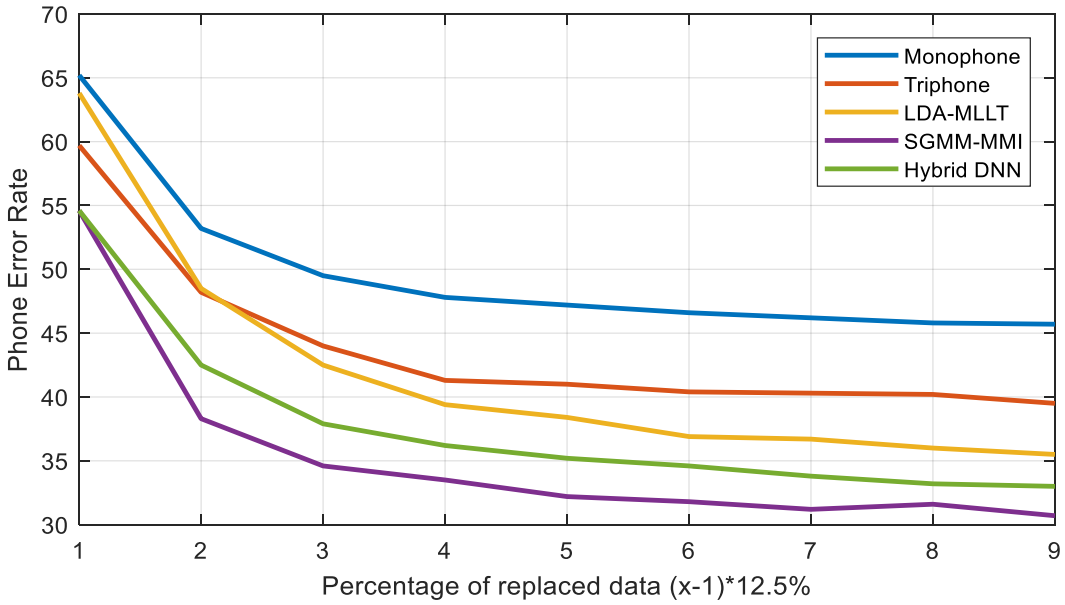


Figure 11.3: PER % vs portions de données perturbées - multiples de 12,5 %

La Figure 11.4 et la Figure 11.5 montrent l'amélioration du PER par rapport à la variation du SNR pour les scénarios HOME et CAR. Il est très intéressant de noter que les tendances générales d'amélioration pour différents modèles avec l'augmentation du SNR sont très similaires à celles de l'expérience présentée dans la Figure 11.3.

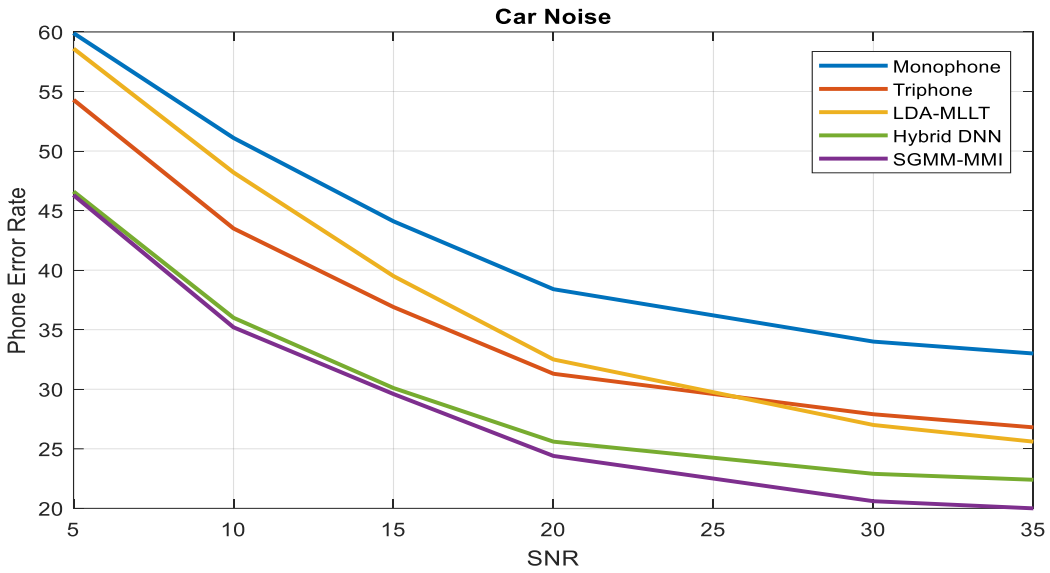


Figure 11.4: Valeurs PER% vs SNR pour NTIMIT dans le scénario CAR

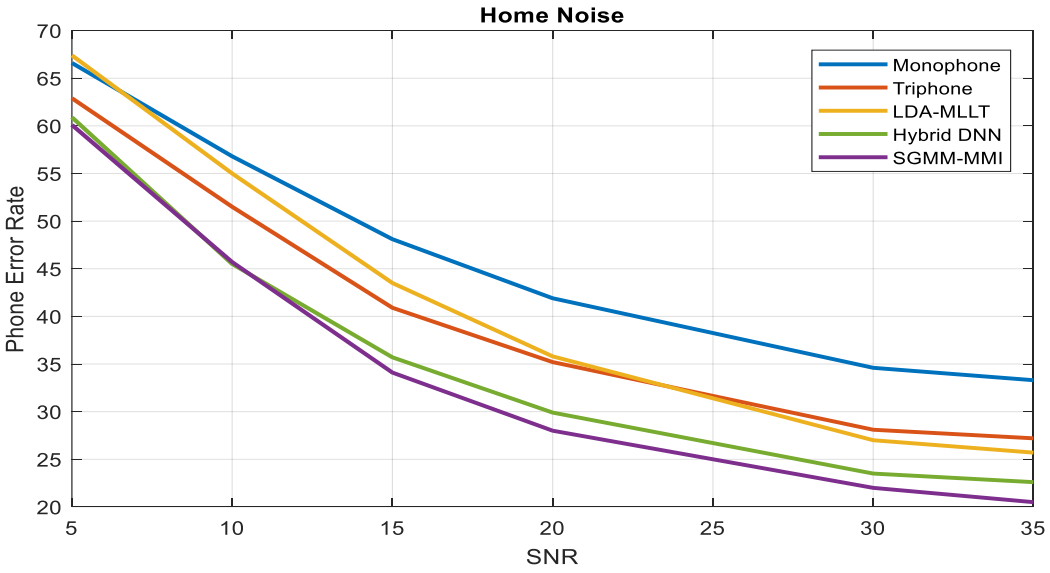


Figure 11.5: Valeurs PER% vs SNR pour NTIMIT dans le scénario HOME

Pour conclure, cette étude approfondit les performances des modèles ASR dans des scénarios réels auxquels sont confrontés les systèmes ASR de nos jours. Cet article étudie quatre vertus principales de certains des systèmes ASR couramment utilisés ; généralisation, adaptation, spécialisation et robustesse. Le pouvoir de généralisation de différents modèles peut avoir un impact important sur les performances du modèle dans de nouvelles applications avec des données de formation limitées dans le nouvel environnement de travail. L'adaptation est un indicateur de l'amélioration du modèle à mesure que de nouvelles données sont ajoutées à l'ensemble d'apprentissage. Les modèles avec transformation ont montré le meilleur pouvoir d'adaptation.

Les modèles monophoniques classiques HMM-GMM s'adaptent de la même manière avec plus de données que les modèles triphones plus complexes. Par conséquent, si les performances initiales sont acceptables, le modèle le plus simple est un meilleur choix car il s'améliorera de la même manière que le modèle triphone plus complexe. Comme mentionné précédemment et comme la plupart des recherches documentaires sont menées, les ensembles de données de référence ont une certaine nature en ce qui concerne les données de formation et de test. De telles restrictions forcent les modèles d'apprentissage à un certain type de discours, et n'évaluent donc pas les performances des modèles pour des scénarios réels où ils seront confrontés à une grande variabilité. Après avoir identifié les capacités des différents modèles dans la thèse, les modèles prometteurs seront fournis avec des jeux de données enrichis synthétiquement, et l'amélioration des performances sera évaluée.

11.7 Chapitre 7 : Augmentation des données de transfert de style neuronal

Inspirée des approches de transfert de style neuronal pour les images, une approche de génération de distorsion synthétique est proposée pour les phonèmes en utilisant le transfert de style des signaux de parole. La Figure 11.6 résume l'idée du transfert de style d'image. L'idée centrale est de capturer les caractéristiques de chaque accent (non natif) à l'aide d'un CNN discriminant, puis d'utiliser la différence entre les caractéristiques extraites par le CNN et celles résultant de l'image actuelle comme fonction de perte. L'approche suppose qu'il y a un signal audio et que seul son accent ou son émotion sera modifié.

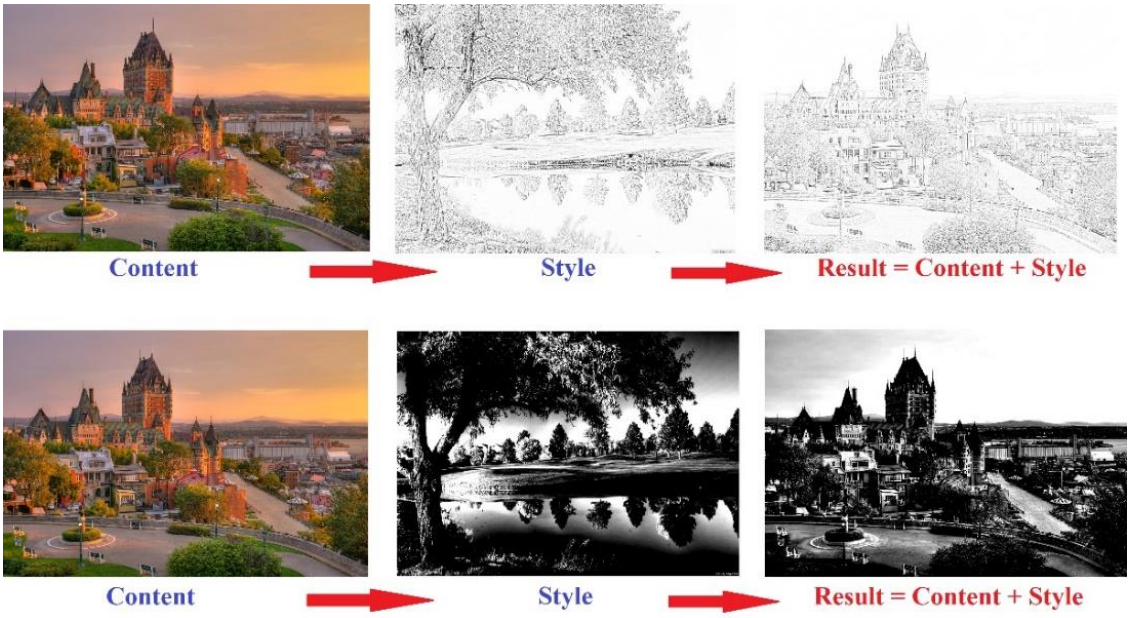


Figure 11.6: Transfert d'images de style neuronal

Nous proposons l'augmentation de style, une nouvelle technique d'augmentation de données basée sur le transfert de style aléatoire, dans le but d'augmenter la résilience des réseaux de neurones convolutifs (CNN) pour les applications de classification et de régression. Nous avons effectué une augmentation de la vitesse, de la prosodie et de l'accent du style pendant l'entraînement, tout en conservant la forme et l'énoncé sémantique.

Ceci est réalisé en adoptant un réseau de transfert de style arbitraire pour exécuter la randomisation de style, au lieu de calculer les intégrations de style cible à partir d'un style audio. Parallèlement aux tests de classification conventionnels, nous examinons l'influence de l'augmentation de style (et de l'augmentation de données en général) sur les tâches de transfert de domaine. Nous démontrons que l'augmentation des données améliore considérablement la résilience au changement de domaine et peut être utilisée à la place de l'adaptation de domaine comme une alternative simple et neutre au domaine.

Nous démontrons que lorsque l'augmentation de style est comparée à une combinaison d'autres approches d'augmentation existantes, elle peut être facilement intégrée à celles-ci pour augmenter les performances du réseau. Nous démontrons l'efficacité de notre technique via des tests de transfert de domaine pour améliorer les performances de divers systèmes de reconnaissance vocale de référence.

11.7.1 Approche proposée

Nous utilisons l'augmentation des données en utilisant un modèle de transfert de style neuronal typique où il y a une observation et une observation de style vers lesquelles nous voulons transformer l'observation. Bien sûr, le transfert de style original fonctionnait avec des images, donc pour utiliser la même stratégie, nous avons dû changer la structure en CNN 1D au lieu de 2D, ce qui est un processus simple. Les entrées diffèrent cependant; nous prenons 3 entrées, l'onde d'origine, les MFCC d'origine et les MFCC d'onde de style. La sortie est le style transféré MFCC.

Ainsi, nous créons directement une observation avec ses MFCC extraits pour être utilisés directement dans le processus de formation. Nous avons également constaté que l'utilisation d'un style d'enceinte proche de celui de l'original améliore les résultats. Par conséquent, nous identifions d'abord les pools de locuteurs à l'aide des vecteurs x, puis nous cartographions de chaque pool de locuteurs à l'autre. Nous utilisons l'ensemble de données TIMIT pour augmenter l'ensemble de données d'apprentissage de l'hispanique-anglais avec diverses proportions.

Pour évaluer les performances de la modélisation acoustique uniquement, les performances sont mesurées par le taux d'erreur phonémique (PER) et non par le taux d'erreur de mots (WER) pour supprimer l'effet du modèle de langue sur les performances du système. Les modèles qui vont être étudiés sont le Gaussian Mixture Model (GMM)-HMM de base, les modèles améliorés utilisant Maximum Likelihood Linear Transform (MLLT) et Linear Discriminant Analysis (LDA) [99], l'hybride HMM – Deep Neural Network (DNN-HMM) [9] et LSTM-HMM [10]. La formation du modèle est effectuée à l'aide de KALDI

11.7.2 Résultats

Le Table 11.10 montre les résultats de l'augmentation des données de style neuronal. Les données synthétisées augmentées par rapport à l'original sont présentées ainsi que le PER correspondant pour les différents modèles. De plus, il est à noter que tous les modèles ont tendance à afficher une baisse de performances lorsque la part de données synthétiques dépasse 50 %. Cela peut être dû à un effet négatif des distorsions dans les données d'apprentissage par rapport aux données d'origine, et le modèle commence à se concentrer sur les distorsions plutôt que de généraliser.

Table 11.10: PER de NST Data Augmentation sur les résultats des tests Hispanic-Eng

Exp	Training Dataset (Percentage)	Model	PER
1	Hispanic-English	GMM-HMM-si	47.3
2		GMM-HMM	40.1
3		DNN-HMM	32.6
4		LSTM-HMM	31.8
5	Hispanic + SynHispanic25	GMM-HMM-si	47.2
6		GMM-HMM	39.7
7		DNN-HMM	28.1
8		LSTM-HMM	26.7
9	Hispanic + SynHispanic50	GMM-HMM-si	46.8
10		GMM-HMM	39.3
11		DNN-HMM	27.5
12		LSTM-HMM	**25.8**
13	Hispanic + SynHispanic75	GMM-HMM-si	47.1
14		GMM-HMM	39.6
15		DNN-HMM	28.0
16		LSTM-HMM	26.1
17	Hispanic + SynHispanic100	GMM-HMM-si	47.4
18		GMM-HMM	40.3
19		DNN-HMM	32.1
20		LSTM-HMM	28.2

11.8 Chapitre 8 : Modèle de génération WaveNet

Wavenet est un réseau autorégressif qui prédit directement une forme d'onde brute échantillon par échantillon. Cette approche a réussi à synthétiser la parole à partir du texte. Cela peut faciliter le processus de génération de la parole synthétique puisqu'il ne nécessite que du texte. Néanmoins, cette approche pourrait nécessiter plus de données pour produire des résultats robustes. Ce sera l'un des points de comparaison lors de la comparaison des différentes approches. Un autre mérite de cette approche est qu'il s'agit d'un modèle unique qui peut être entraîné sur tous les types de variabilité, et simplement conditionné sur un ou plusieurs de ces types lors de la génération. Ce modèle est également basé sur les CNN, mais ses couches convolutionnelles ne convoluent pas des fenêtres contiguës d'échantillons, mais convoluent plutôt tous les autres échantillons pour capturer les dépendances à long terme avec moins de

calculs. On parle alors de circonvolution dilatée. Dans cette étude, nous présentons WaveNet-DA, une technique d'augmentation de données basée sur le modèle génératif WaveNet. Voici quelques-uns des avantages de notre méthode.

- Premièrement, nous utilisons un modèle génératif dans notre méthode. Les approches précédentes se sont concentrées sur les changements de fonctionnalités. Cependant, l'inconvénient de la modification des caractéristiques est que le modèle acoustique (AM) peut être en mesure de faire la différence entre les données d'origine et les données modifiées. Les données améliorées ne sont plus une nouvelle représentation une fois que le modèle acoustique comprend la connexion, et ainsi sa valeur est diminuée.
- Deuxièmement, au lieu d'utiliser des vocodeurs traditionnels, nous utilisons WaveNet pour produire des énoncés. Pendant le paramétrage, les vocodeurs traditionnels perdent des informations détaillées, ce qui entraîne des artefacts dans la parole de sortie. Ces artefacts, qui ne sont pas présents dans les données réelles, pourraient rendre difficile la généralisation de l'AM.
- Enfin, notre méthode produit de la parole avec une variété de modèles de hauteur. Les paramètres du vocodeur comme la fréquence fondamentale et les informations spectrales n'ont aucun effet sur notre WaveNet-DA. En conséquence, étant donné que WaveNet ne dispose pas de détails spécifiques sur les propriétés acoustiques telles que la variabilité de hauteur, la parole synthétique peut avoir une variété de modèles de hauteur. Nous démontrons que l'approche suggérée surpasse DA en utilisant la perturbation de vitesse, qui est la DA la plus réussie à ce jour, sur le corpus hispano-anglais.

Nous avons utilisé 64 piles de couches résiduelles avec des convolutions dilatées de taille 25 ms et une perte d'entropie croisée pour former DNN-HMM AM. Des triphones à alignements forcés ont été utilisés pour l'entraîner à l'aide de GMM-HMM AM préalablement entraîné. Sa caractéristique d'entrée est une banque de filtres avec 40 coefficients d'échelle Mel plus la valeur d'énergie et les première et seconde dérivées temporelles, ce qui donne un vecteur d'entrée de 123 dimensions à chaque image. Nous avons utilisé une fenêtre de Hamming de 25 msec toutes les 10 msec pour la fonction de banque de filtres, et chaque caractéristique d'entrée a été normalisée pour chaque syllabe unique. Un modèle de langage trigramme élagué a été utilisé avec une taille de faisceau de 10 pendant le décodage.

11.8.1 Résultats

WaveNet-DA montre une amélioration plus élevée pour tous les modèles ASR, comme indiqué dans le Table 11.11 avec une augmentation des données de 50 %, mais bien sûr, le montant de l'augmentation diffère d'un modèle à l'autre en fonction de sa capacité. Plus le modèle a de capacité, plus il s'améliore. Il peut sembler inattendu que le DNN obtienne des performances moindres que LSTM, mais comme LSTM a une manière plus contrainte de lier les poids et d'établir des relations entre les observations dans le temps, il bénéficie plus que le DNN entièrement connecté.

Table 11.11: Augmentation WaveNet vs perturbation de la vitesse

Augmentation	Model	PER
Hispanic + SynWaveNet	GMM-HMM-si	41.3
	GMM-HMM	34.2
	GMM-HMM+DNN	23.3
	GMM-HMM+LSTM	**21.9**
Hispanic + Perturbation	GMM-HMM-si	42
	GMM-HMM	37.1
	GMM-HMM+DNN	29.3
	GMM-HMM+LSTM	28.4

Contrairement aux scénarios VC précédents, notre WaveNet crée un discours ressemblant au locuteur cible avec variété car il est conditionné localement exclusivement sur des informations linguistiques et énergétiques. Il a montré qu'il améliore le PER de l'hispano-anglais en utilisant la parole synthétisée à partir du corpus TIMIT plus que la perturbation de vitesse, qui est l'approche DA connue la plus efficace.

11.9 Chapitre 9 : Auto-encodeur récurrent

Une approche AE récurrente (RAE) est proposée pour être entraînée sur une parole non native ou accentuée et conditionnée sur un certain phonème à l'aide d'un réseau à entrées multiples. La raison du conditionnement est que certains accents non natifs prononcent par erreur certains phonèmes avec d'autres. Si le réseau est laissé libre d'encoder le discours natif vers son homologue non natif, le réseau ne tiendra pas compte de ces défauts de prononciation et choisira simplement le phonème le plus proche du natif.

Cette étude démontre l'utilité de l'augmentation des données en utilisant un cadre RVAE et l'utilité des caractéristiques basées sur les variables latentes dans l'ASR. La modélisation acoustique nécessite un corpus de parole (données audio d'entraînement) avec transcription de phonèmes (ou de syllabes) pour l'ASR. À mesure que le nombre d'époques d'entraînement augmente, les performances de l'ASR s'améliorent. Cependant, la préparation d'un corpus vocal contenant des étiquettes de phonèmes pour la formation de modèles acoustiques pour des applications particulières, telles que la reconnaissance vocale senior et l'ASR avec des langues à faibles ressources, est un défi. Dans de tels cas, il est impossible de créer suffisamment d'époques d'entraînement pour la modélisation acoustique, et par conséquent suffisamment d'ASR. La performance ne peut pas être acquise.

Un RVAE est composé de deux composants : un encodeur qui extrait un vecteur latent des variables d'entrée (observation) et un décodeur qui reconstruit les variables d'origine à partir du vecteur latent. Un vecteur latent contient des "significations", qui sont des représentations abstraites des variables d'entrée, à partir desquelles un décodeur peut reconstruire les variables d'entrée. Ce travail fournit une stratégie pour augmenter les données à l'aide d'un RVAE. Une forme d'onde vocale est formée en codant un vecteur latent à partir d'une forme d'onde d'entrée dans un vecteur latent. La forme d'onde produite a le même contenu et la même durée que la forme d'onde d'entrée

11.9.1 Approche proposée

Pour minimiser l'écart entre les données d'entrée et reproduites, les locuteurs sont d'abord regroupés sur la base des caractéristiques du vecteur x. Ensuite, chaque audio est découpé en segments superposés de 0,5 seconde chacun avec un chevauchement de 50 % pour minimiser les distorsions lors de la reconstruction.

Le RVAE est ensuite formé sur un ensemble donné, et à partir du nouveau domaine, les locuteurs sont ensuite affectés à leurs clusters appropriés et de nouvelles données sont reconstruites via le RVAE précédemment formé. Ce qui est attendu, c'est que le RVAE ne peut générer que des données similaires à ce qu'il a appris précédemment. Ainsi, on s'attend à ce que les données générées à partir d'un nouveau type de parole différent soient mappées sur les mêmes caractéristiques de l'ensemble d'apprentissage d'origine.

11.9.2 Résultats

Les PER obtenus à l'aide des modèles acoustiques entraînés avec la parole hispano-anglaise et ses versions augmentées sont représentés dans le tableau 9.1. LSTM-HMM montre la meilleure amélioration lorsque l'augmentation des données est utilisée. Comme démontré, Data Augmentation a un effet marginal sur le PER du modèle GMM-HMM (tri2, tri3 et tri4) par rapport aux modèles entraînés à l'aide de l'ensemble de données hispanique-anglais seul. Comme observé dans le tableau 9.1, les PER du discours reconstruit sont supérieurs de quelques points de pourcentage à ceux des discours originaux ; par conséquent, DA via le RVAE n'a eu aucun effet sur le raffinement des modèles basés sur GMM. D'autre part, le modèle DNN formé à l'aide de l'ensemble de données hispaniques amélioré a considérablement amélioré le PER pour (Hispanic-Eng + timit_reconstructed25) et (Hispanic-Eng +timit_reconstructed50). En conséquence, la DA suggérée a été efficace pour générer un modèle acoustique précis. Avec TIMIT, l'augmentation a entraîné une amélioration relative de 4,5 %.

Table 11.12: Test TIMIT utilisant l'hispanique-eng reconstruit. données d'un TIMIT RVAE

Training Dataset (Percentage)	Model	PER
Hispanic-English	GMM-HMM-si	47.3
	GMM-HMM	40.1
	DNN-HMM	32.6
	LSTM-HMM	31.8
Hispanic-Eng. + timit_reconstructed25	GMM-HMM-si	46.3
	GMM-HMM	38.9
	DNN-HMM	27.1
	LSTM-HMM	25.3
Hispanic-Eng. + timit_reconstructed50	GMM-HMM-si	46.8
	GMM-HMM	39.3
	DNN-HMM	27.5
	LSTM-HMM	25.8
Hispanic-Eng. + timit_reconstructed75	GMM-HMM-si	46.6
	GMM-HMM	39
	DNN-HMM	27.1
	LSTM-HMM	25.1
Hispanic-Eng. + timit_reconstructed100	GMM-HMM-si	46.6
	GMM-HMM	39.4
	DNN-HMM	27
	LSTM-HMM	25.3

Il est également à noter que l'augmentation des données RVAE ne détériore pas les performances globales comme les autres techniques proposées. Cela peut être dû à la nature de la reconstruction elle-même qui dépend d'un domaine de cartographie plutôt que de mélanger plus d'une fonction de cartographie.

11.10 Chapitre 10 : Conclusion et contributions

Cette thèse étudie la capacité de différents modèles ASR à s'adapter aux variabilités de la parole. Le premier type de flexibilité provient du modèle d'apprentissage lui-même. Pourtant, il est très difficile à contrôler car incorporer plus de variabilités signifie nécessiter plus de capacité de modélisation qui à son tour nécessite plus de données. Bien sûr, en raison de l'expansion des applications ASR, ainsi que des préoccupations soulevées concernant la confidentialité des données, la disponibilité de plus de données devient un défi. Par conséquent, cette thèse a proposé trois nouvelles techniques d'augmentation de données qui n'ajoutent pas seulement des variations de propriétés, mais aussi de structure. Le premier est Neural Style Transfer, où les observations d'un domaine sont imitées par un autre domaine produisant des résultats intéressants.

Ce type d'augmentation est très efficace, mais exigeant en termes de calcul. La seconde est la modélisation générative WaveNet. Il était utilisé auparavant pour TTS, mais ici il est étudié pour l'augmentation des données. Ce type n'a besoin d'être formé que sur le discours cible, mais encore une fois, car lui-même a besoin de données à former qui peuvent devenir un obstacle à l'utilisation de cette méthode. Ceci est recommandé lorsque le nouveau domaine est proche de l'original. Enfin, l'auto-encodeur récurrent se situe quelque part entre les deux approches en termes d'exigences de calcul et de qualité des résultats.

Appendix A

Experimentation Challenges

Throughout the experimental work of this thesis, there has been difficulties that faced the production of reasonable results. This of course happens with any type of experimental work, and it is good practice to share those difficulties and notes so that future work can avoid similar mistakes, producing more fruitful experiments. In this appendix we categorize those challenges into three categories: data preparation, model architecture, and result interpretation.

A1 Data Preparation

Cleaning and preparing any speech data is a challenge of its own, and in this thesis, data is the most important component. Since the goal of this work is proposing new data augmentation techniques, then extra care should be given to the data mixing and data cohesion before going into the modelling phase. Working with multiple domains and trying to match one domain to another was the first challenge. Speech synthesis models encountered less difficulties in this case since the creation technique was essentially new, nevertheless, trying to match a target speaker from a different dataset was challenging. At first, the results were a complete failure. Then, the problems of mismatch started to unravel. The following reasons were identified:

1- Since WaveNet works on the time domain, the sampling rate must be matched for both datasets. This had to be unified before any training process was done.

2- Working with Style Transfer on the other hand, showed another type of mismatch, which is the speaker. Although the essence is to mimic the style of one speaker to another, the model is mainly trained on a dataset of certain type of speaking style and a certain domain. When this is very deviant from the style speaker, the results come out poor. In order to mitigate this, pools of speakers were created for both source speakers and target speaker would be matched to a fine tuned model of Neural Style Transfer on the pool that matches the style speaker.

3- Finally, the Recurrent Variational Autoencoder posed yet another challenge which is duration encoding. Attempting to reconstruct complete audio files was a complete failure. To overcome this challenge, the reconstruction was first altered to work on the MFCC domain which has lesser redundancy that the audio itself and lesser length to encode. Moreover, the reconstruction was not done on the whole file, but rather reconstructing chunk by chunk. Since the aim is data augmentation and not the actual reconstruction, worrying about the transitions in features of each chunk was not a concern. The reconstructed features would then be introduced to the model.

A2 Model Architecture

Although neural networks showed great success in numerous applications, it stills suffers from several challenges when it comes to design and interpretation. In this thesis we are more concerned about the design as interpretation is out of scope.

Choosing the appropriate objective function, and number of neurons or layers are some of the major challenges. Using the original models, as is, rarely works on new applications as is. The reason again is the difference in domain and type of data. What we came to notice after running the experiments is the following:

1- The objective function is the most impacting choice on the results. For the purpose of data augmentation which is neither classification nor reconstruction, the Kullback Leibler Divergence is the most tolerable objective function as it seeks a matching distribution rather than an exact match. Accordingly, similar divergence objective functions can be investigated in the future in similar research topics.

2- The number of neurons and layers do impact the results, however, not as significant as the objective function. Nevertheless, the impact on computations is significant. Trying to achieve higher accuracy by adding more neurons and layers works, but the computations start to hinder the practicality. The work around here was focusing on regularization techniques that seek reducing the weights or penalizing them, for example, dropout and L_1 norms. Later after training is complete, some neurons can be taken out safely during the testing process without much impact on the results.

A3 Result Interpretation

The improvements when investigating data augmentation techniques can seem straight forward, yet the reasoning behind the improvement can be challenging. The issues faced here came from the fact that in some experiments the reconstruction measures seemed reasonable, however, the ASR improvements were deviant, and vice versa.

This problem became a bit clearer when the Kullback Leibler Divergence was utilized, yet not clear enough. Still there were some unexplained deviations between the results of the augmentation quality and the performance of the ASR performance.

The first reason which made sense was the fact that both performance measures are not tied together. Actually, they are two completely different problems, so there is no mathematical guarantee that the optimality in one of them will improve the other. This actually opens the potential for future research to investigate establishing this tie when it comes to data augmentation.

Second, we noticed that good reconstruction performance can mean more overfitting to the data and hance does not generalize well when attempting to transfer to another domain. Utilizing perturbation methods for neural networks, such as adding weight noise, improved interpretation of the results, establishing a better link between both the performance of the ASR system and the data augmentation system.

See Table - Appendix A 1 and Table - Appendix A 2 below from our batch of failed experiments.

Table - Appendix A 1: PER results of NST Data Augmentation on Hispanic-English test results

Exp	Training Dataset (Percentage)	Model	PER
1	Hispanic-English	GMM-HMM-si	91.77
2		GMM-HMM	94.46
3		GMM-HMM+DNN	53.88
4		GMM-HMM+LSTM	51.14
5	Hispanic + SynHispanic25	GMM-HMM-si	88.15
6		GMM-HMM	91.7
7		GMM-HMM+DNN	53.75
8		GMM-HMM+LSTM	51.01
9	Hispanic + SynHispanic50	GMM-HMM-si	91.34
10		GMM-HMM	93.87
11		GMM-HMM+DNN	52.88
12		GMM-HMM+LSTM	**50.76**
13	Hispanic + SynHispanic75	GMM-HMM-si	96.43
14		GMM-HMM	90.22
15		GMM-HMM+DNN	97.22
16		GMM-HMM+LSTM	90.46
17	Hispanic + SynHispanic100	GMM-HMM-si	97.07
18		GMM-HMM	97.71
19		GMM-HMM+DNN	90.95
20		GMM-HMM+LSTM	91.26
21	Hispanic + SynHispanic100 + SynTimit25	GMM-HMM-si	98.9
22		GMM-HMM	99.12
23		GMM-HMM+DNN	91.37
24		GMM-HMM+LSTM	91.85
25	Hispanic + SynHispanic100 + SynTimit50	GMM-HMM-si	99.16
26		GMM-HMM	99.3
27		GMM-HMM+DNN	92.34
28		GMM-HMM+LSTM	92.46
29	Hispanic + SynHispanic100 + SynTimit75	GMM-HMM-si	99.17
30		GMM-HMM	99.24
31		GMM-HMM+DNN	92.64
32		GMM-HMM+LSTM	92.74
33	Hisp-Eng + SynHispanic100 + SynTimit100	GMM-HMM-si	99.6
34		GMM-HMM	99.46
35		GMM-HMM+DNN	94.65
36		GMM-HMM+LSTM	94.23

Table - Appendix A 2: PER results of NST Data Augmentation on TIMIT test results

Exp	Training Dataset (Percentage)	Model	PER
1	Hispanic-English	GMM-HMM-si	96.75
2		GMM-HMM	96.75
3		GMM-HMM+DNN	55.58
4		GMM-HMM+LSTM	55.52
5	Hispanic + SynHispanic25	GMM-HMM-si	92.45
6		GMM-HMM	92.8
7		GMM-HMM+DNN	**48.33**
8		GMM-HMM+LSTM	48.91
9	Hispanic + SynHispanic50	GMM-HMM-si	95.29
10		GMM-HMM	95.34
11		GMM-HMM+DNN	48.64
12		GMM-HMM+LSTM	49.15
13	Hispanic + SynHispanic75	GMM-HMM-si	97.96
14		GMM-HMM	97.95
15		GMM-HMM+DNN	89.3
16		GMM-HMM+LSTM	90.38
17	Hispanic + SynHispanic100	GMM-HMM-si	97.03
18		GMM-HMM	97.05
19		GMM-HMM+DNN	91.72
20		GMM-HMM+LSTM	91.89
21	Hispanic + SynHispanic100 + SynTimit25	GMM-HMM-si	98.25
22		GMM-HMM	98.26
23		GMM-HMM+DNN	89.59
24		GMM-HMM+LSTM	90.34
25	Hispanic + SynHispanic100 + SynTimit50	GMM-HMM-si	99.03
26		GMM-HMM	99.03
27		GMM-HMM+DNN	91.4
28		GMM-HMM+LSTM	90.91
29	Hispanic + SynHispanic100 + SynTimit75	GMM-HMM-si	98.66
30		GMM-HMM	98.66
31		GMM-HMM+DNN	92.68
32		GMM-HMM+LSTM	92.64
33	Hispanic + SynHispanic100 + SynTimit100	GMM-HMM-si	99.07
34		GMM-HMM	99.07
35		GMM-HMM+DNN	94.66
36		GMM-HMM+LSTM	94.63

Appendix B

Kaldi Toolkit

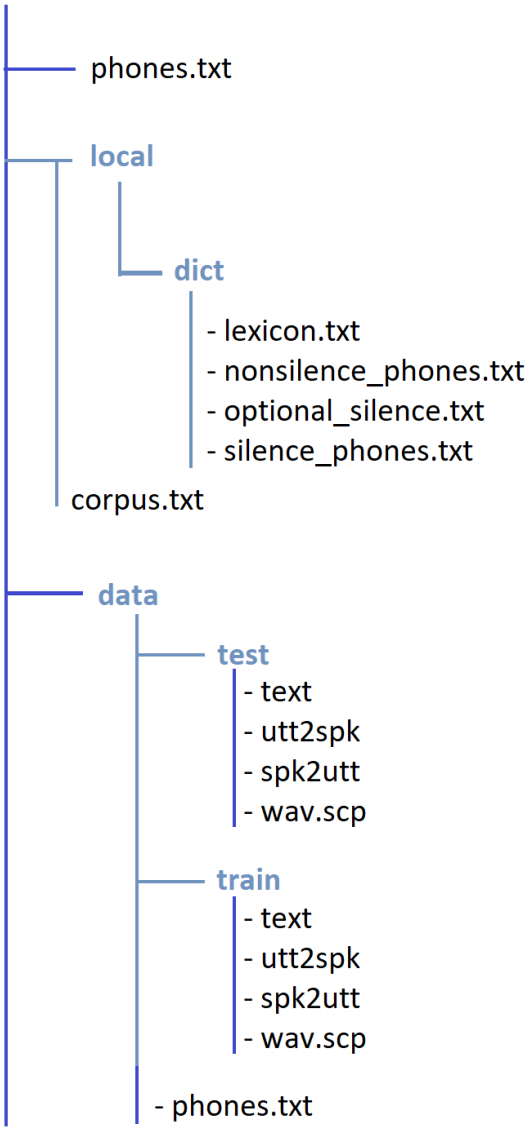
Kaldi is a well-documented free and open-source toolset for academics working on speech recognition. The toolkit originated during a 2009 Johns Hopkins University workshop dubbed "Low Development Cost, High-Quality Speech Recognition for New Languages and Domains." Initially, the Kaldi project focused on modelling using Subspace Gaussian Mixture Models (SGMMs) and some research on lexicon learning; from there, the Kaldi project evolved into what it is today.

Kaldi is written in C++ and has extensive support for linear algebra, as well as the ability to produce features such as MFCCs, fbank, and fMLLR. As a result, Kaldi is often utilized in current research on deep neural networks to pre-process raw waveforms into acoustic features for end-to-end neural models. Linux is the optimal environment for Kaldi. For our work, we chose Ubuntu 18.04, a feature-rich and reliable Linux distribution.

B1 Data Preparation

The standard audio format used in Kaldi is WAV. The following files and directories need to be created as demonstrated in the tree Appendix - Figure B 1.

- Spk2gender: a record of the speakers gender
- wav.scp: connects each utterance to an audio file
- text: connects each utterance with its transcription
- utt2spk: identifies which utterance belongs to which speaker
- corpus.txt: contains the transcription of all utterances
- lexicon.txt: contains every word from our experiment dictionary with phone transcriptions
- nonsilence_phones.txt: contains nonsilence phones present in our experiment
- silence_phones.txt: contains silence phones
- optional_silence.txt: contains optional silence phones



Appendix - Figure B 1: Kaldi Data Preparation Structure

B2 Running Scripts

Kaldi's run.sh script executes all process stages, including data preparation, feature extraction, training, and decoding. Table - Appendix B 1 demonstrates the run.sh script, which gives insight into the overall Kaldi technique:

Table - Appendix B 1: run.sh script in Kaldi

```
#!/usr/bin/env bash

# Data Training

# 1: Feature extraction (MFCC & CMVN)
# 2: Preparing Language Model (Language Data & G.fst (Grammar Lang model
file))
# 3: Monophone training
# 4: Monophone alignment (used for Triphone training preparation)
# 5: Triphone training (Tri1, Tri2, Tri3, ...etc) see link below
    # http://jrmeyer.github.io/asr/2019/08/17/Kaldi-cheatsheet.html
# 6: Triphone alignment
# 7: Decoding

stage=0 # This is for a choice of one of the above stages

mfccdir=mfcc
nj=8
. utils/parse_options.sh
. ./cmd.sh
. ./path.sh
set -e


# Stage 1 (Feature Extraction)
echo $stage $nj
if [ $stage -eq 1 ]; then
  echo "Generating features for training set"
  steps/make_mfcc.sh --cmd "$train_cmd" --nj $nj data/train
exp/make_mfcc/train $mfccdir
  steps/compute_cmvn_stats.sh data/train exp/make_mfcc/train $mfccdir

  #echo "Generating features for test set"
  #steps/make_mfcc.sh --cmd "$train_cmd" --nj $nj data/test
    #exp/make_mfcc/test $mfccdir
  #steps/compute_cmvn_stats.sh data/test exp/make_mfcc/test $mfccdir
  # after training, we can test any set of audio. Just change "train" to
    #"test" lines 29-31

fi


# Stage 2 (LM)
if [ $stage -eq 2 ]; then
  echo "Preparing Language data. Creates files in data/local"
  utils/prepare_lang.sh data/local/dict "<UNK>" data/local/lang data/lang
  echo
  echo "===== LANGUAGE MODEL CREATION ====="
  echo "===== MAKING lm.arpa ====="
  echo
```

```
  loc=`which ngram-count`;
  if [ -z $loc ]; then
          if uname -a | grep 64 >/dev/null; then
                  sdir=$KALDI_ROOT/tools/srilm/bin/i686-m64
          else
                          sdir=$KALDI_ROOT/tools/srilm/bin/i686
          fi
          if [ -f $sdir/ngram-count ]; then
                          echo "Using SRILM language modelling tool from
$sdir"
                          export PATH=$PATH:$sdir
          else
                          echo "SRILM toolkit is probably not installed.
                                  Instructions: tools/install_srilm.sh"
                          exit 1
          fi
  fi
  local=data/local
  mkdir $local/tmp
  ngram-count -order $lm_order -write-vocab $local/tmp/vocab-full.txt -
wbdiscount -text $local/corpus.txt -lm $local/tmp/lm.arpa
  echo
  echo "===== MAKING G.fst ====="
  echo
  lang=data/lang
  arpa2fst --disambig-symbol=#0 --read-symbol-table=$lang/words.txt
$local/tmp/lm.arpa $lang/G.fst
fi


# Stage 3 (Mono Training)
if [ $stage -eq 3 ]; then
  echo "===== MONO TRAINING ====="
  echo
  steps/train_mono.sh --nj $nj --cmd "$train_cmd" data/train data/lang
exp/mono
fi


# Stage 4: Monophone alignment (mapping the phones to frames)
if [ $stage -eq 4 ]; then
  steps/align_si.sh --nj $nj --cmd "$train_cmd" data/train data/lang
exp/mono exp/mono_align
fi


# Stage 5: Triphone-1 TRAINING
if [ $stage -eq 5 ]; then
  echo " === Training Tri1 model. Features are delta + delta-delta"
  steps/train_deltas.sh --cmd "$train_cmd" 3000 30000 data/train
data/lang exp/mono_align exp/tri1
fi
```

```
103
104 # Stage 6: Triphone-1 alignment
105 if [ $stage -eq 6 ]; then
106   echo "==== Aligning Tri1 Model ===="
107   steps/align_si.sh --nj $nj --cmd "$train_cmd" data/train data/lang
108 exp/tri1 exp/tri1_align
109 fi
110
111
112 # Stage 7: Triphone-2 TRAINING
113 if [ $stage -eq 7 ]; then
114   echo " === Training Tri2 model. Features are LDA + MLLT"
115   steps/train_lda_mllt.sh --cmd "$train_cmd" 5000 50000 data/train
116 data/lang exp/tri1_align exp/tri2
117 fi
118
119
120 # Stage 8: Triphone-2 alignment
121 if [ $stage -eq 8 ]; then
122   echo "==== Aligning Tri2 Model ===="
123   steps/align_si.sh --nj $nj --cmd "$train_cmd" data/train data/lang
124 exp/tri2 exp/tri2_align
125 fi
126
127
128 # Stage 9: Triphone-3 TRAINING
129 if [ $stage -eq 9 ]; then
130   echo " === Training Tri3 model. Features are LDA + MLLT + SAT"
131   steps/train_sat.sh --cmd "$train_cmd" 8000 150000 data/train data/lang
132 exp/tri2_align exp/tri3
133 fi
134
135
136 # Stage 10: Triphone-3 alignment
137 if [ $stage -eq 10 ]; then
138   echo "==== Aligning Tri3 Model ===="
139   steps/align_fmllr.sh --nj $nj --cmd "$train_cmd" data/train data/lang
140 exp/tri3 exp/tri3_align
141 fi
142
    #================================================================


    # Stage 11: Decoding
    if [ $stage -eq 11 ]; then
      echo "===== MONO DECODING ====="
      echo
      utils/mkgraph.sh --mono data/lang exp/mono exp/mono/graph || exit 1
      steps/decode.sh --nj $nj --cmd "$decode_cmd" exp/mono/graph data/train
    exp/mono/decode
```

```
  echo "===== TRI1 (first triphone pass) DECODING ====="
  echo
  utils/mkgraph.sh data/lang exp/tri1 exp/tri1/graph || exit 1
  steps/decode.sh --nj $nj --cmd "$decode_cmd" exp/tri1/graph data/test
exp/tri1/decode


  echo "===== TRI2 (first triphone pass) DECODING ====="
  echo
  utils/mkgraph.sh data/lang exp/tri2 exp/tri2/graph || exit 1
  steps/decode.sh --nj $nj --cmd "$decode_cmd" exp/tri2/graph data/test
exp/tri2/decode


  echo "===== TRI3 (first triphone pass) DECODING ====="
  echo
  utils/mkgraph.sh data/lang exp/tri3 exp/tri3/graph || exit 1
  steps/decode_fmllr.sh --nj $nj --cmd "$decode_cmd" exp/tri3/graph
data/test exp/tri3/decode
fi
```

B3 Script Steps

- Stage 1 : This stage is the feature extraction stage where MFCCs are created (script lines 25-40).
- Stage 2 : The Language Model (LM) is created (script lines 43-78).
- Stage 3 : Monophone training with fixed 2000 Gaussians as default (script lines 81-87).
- Stage 4 : Monophone alignment. Mapping phones to frames (script lines 90-94).
- Stage 5 : Triphone-1 training. Features are delta + delta-delta. Number of leaves range of selection is 2,000 to 5,000. Total gaussian range of selection is 10,000 to 50,000 (script lines 97-102).
- Stage 6 : Triphone-1 alignment (script lines 104-109).
- Stage 7 : Triphone-2 training. Features are LDA + MLLT. Number of leaves range of selection is 2,500 to 7,500. Total gaussian range of selection is 15,000 to 75,000 (script lines 112-117).
- Stage 8 : Triphone-2 alignment (script lines 120-125).
- Stage 9 : Triphone-3 training. Features are LDA + MLLT + SAT. Number of leaves range of selection is 2,500 to 10,000. Total gaussian range of selection is 15,000 to 200,000 (script lines 128-133).

- Stage 10 : Triphone-3 alignment (script lines 136-141).
- Stage 11 : Mono, Tri-1, Tri-2, and Tri-3 decoding (script lines 142 to end of script)
- Stage NN : This is the stage where we move forward with neural network training, after the Triphone-3 alignments are generated.

References

[1] Z. Zhang, J. Geiger, J. Pohjalainen, A. E.-D. Mousa, W. Jin, and B. Schuller, "Deep Learning for Environmentally Robust Speech Recognition: An Overview of Recent Developments," May 2017, [Online]. Available: http://arxiv.org/abs/1705.10874

[2] Robert Delaney, "Dialect Map of American English," Apr. 15, 2013. http://robertspage.com/dialects.html (accessed Apr. 08, 2022).

[3] K. Jambrosic, M. Horvat, and H. Domitrovic, "Reverberation time measuring methods," in *Proceedings - European Conference on Noise Control*, 2008, pp. 4503–4508. doi: 10.1121/1.2934829.

[4] Seltzer and Michael L., "Microphone Array Processing for Robust Speech Recognition," Carnegie Mellon University, 2003.

[5] D. Bagchi, S. Wotherspoon, Z. Jiang, and P. Muthukumar, "Speech Synthesis as Augmentation for Low-Resource ASR," Dec. 2020, [Online]. Available: http://arxiv.org/abs/2012.13004

[6] Y. Rebryk and S. Beliaev, "ConVoice: Real-Time Zero-Shot Voice Style Transfer with Convolutional Network," 2020. [Online]. Available: https://github.com/CorentinJ/

[7] S. J. Cheon, J. Y. Lee, B. J. Choi, H. Lee, and N. S. Kim, "Gated recurrent attention for multi-style speech synthesis," *Applied Sciences (Switzerland)*, vol. 10, no. 15, Jul. 2020, doi: 10.3390/APP10155325.

[8] X. An, F. K. Soong, and L. Xie, "Improving Performance of Seen and Unseen Speech Style Transfer in End-to-end Neural TTS," 2021. [Online]. Available: https://xiaochunan.github.

[9] G. Hinton *et al.*, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," *IEEE Signal Processing Magazine*, no. November, pp. 82–97, 2012. doi: 10.1109/MSP.2012.2205597.

[10] D. Povey *et al.*, "The subspace Gaussian mixture model—A structured model for speech recognition," *Computer Speech & Language*, vol. 25, no. 2, pp. 404–439, 2011, doi: 10.1016/j.csl.2010.06.003.

[11] I. Sutskever, O. Vinyals, and Q. V Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 3104–3112.

[12] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, “Connectionist Temporal Classification : Labelling Unsegmented Sequence Data with Recurrent Neural Networks,” in *Proceedings of the 23rd international conference on Machine Learning*, 2006, pp. 369–376. doi: 10.1145/1143844.1143891.

[13] Milton Orlando Sarria Paja, “TOWARDS MULTIPLE VOCAL EFFORT SPEAKER VERIFICATION: EXPLORING SPEAKER-DEPENDENT INVARIANT INFORMATION BETWEEN NORMAL AND WHISPERED SPEECH,” PhD dissertation, Institut national de la recherche scientifique, Montreal, 2016.

[14] D. O’Shaughnessy, *Speech Communications Human and Machine*, Second. IEEE Press, 2000.

[15] H. Fayek, “Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What’s In-Between,” 2016. https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html

[16] E. Wong and S. Sridharan, “Comparison of linear prediction cepstrum coefficients and mel-frequency cepstrum coefficients for language identification,” *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing, ISIMP 2001*, pp. 95–98, Feb. 2001, doi: 10.1109/ISIMP.2001.925340.

[17] S. Young, G. Evermann, and et al, *The HTK Book*. 1995.

[18] W. Abdulla and N. Kasabov, “The Concepts of Hidden Markov Model in Speech Recognition,” 1999, [Online]. Available: www:http://www.otago.ac.nz/informationscience/

[19] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” *Proceedings of the IEEE*, vol. 77, no. 2. pp. 257–286, 1989. doi: 10.1109/5.18626.

[20] X. Chen, “Scalable Recurrent Neural Network Language Models for Speech Recognition,” PhD dissertation, Cambridge University, 2017.

[21] J. Du, Y. Hu, and H. Jiang, “Boosted Mixture Learning of Gaussian Mixture HMMs for Speech Recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2091–2100, 2011, doi: doi: 10.1109/TASL.2011.2112352.

[22] Heigold and Georg & Schlüter, “Discriminative Training for Automatic Speech Recognition: Modeling, Criteria, Optimization, Implementation, and Performance.,” *IEEE Signal Processing Magazine.*, vol. 29, no. 10.1109/MSP.2012.2197232, 2012.

[23] Douglas A. Reynolds, “GAUSSIAN MIXTURE MODELS,” *Encyclopedia of Biometrics*, pp. 659–663, 2009.

[24] D. Yu and L. Deng, *Automatic Speech Recognition - A Deep Learning Approach*. London: Springer, 2015.

[25] D. Yu and M. L. Seltzer, “Feature Learning in Deep Neural Networks - Studies on Speech Recognition Tasks,” Jan. 2013, [Online]. Available: http://arxiv.org/abs/1301.3605

[26] J. Dean and G. Corrado, “Large scale distributed deep networks,” *Adv. Neural Inf. Process. Syst.*, 2012.

[27] G. E. Dahl, T. N. Sainath, and G. E. Hinton, “IMPROVING DEEP NEURAL NETWORKS FOR LVCSR USING RECTIFIED LINEAR UNITS AND DROPOUT.”

[28] N. Srivastava and G. Hinton, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[29] X. L. Zhang and D. L. Wang, “Boosting contextual information for deep neural network based voice activity detection,” *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 24, no. 2, pp. 252–264, Feb. 2016, doi: 10.1109/TASLP.2015.2505415.

[30] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” University of Toronto.

[31] L. Li *et al.*, “Hybrid Deep Neural Network–Hidden Markov Model (DNN-HMM) Based Speech Emotion Recognition,” in *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, 2013, pp. 312–317. doi: 10.1109/ACII.2013.58.

[32] C. Cai and Y. Xu, “Deep Neural Networks with Multistate Activation Functions,” vol. 2015, 2015, doi: 10.1155/2015/721367.

[33] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6. doi: 10.1109/ICEngTechnol.2017.8308186.

[34] Jui-Ting Huang, J. Li, and Y. Gong, “AN ANALYSIS OF CONVOLUTIONAL NEURAL NETWORKS FOR SPEECH RECOGNITION, 2015 IEEE International Conference on Acoustics, Speech and Signal Processing.”

[35] Prateek Verma and Julius Smith, “Neural Style Transfer for Audio Spectrograms,” *arXiv:1801.01589*, vol. 1, Jan. 2018, Accessed: Apr. 21, 2022. [Online]. Available: https://arxiv.org/pdf/1801.01589.pdf

[36] Y. Jing, Y. Yang, and et al, “Neural Style Transfer: A Review,” *arXiv:1705.04058*, vol. 7, Oct. 2018, [Online]. Available: http://arxiv.org/abs/1705.04058

[37] Lonce Wyse, “Audio spectrogram representations for processing with Convolutional Neural Networks,” May 2017. Accessed: Apr. 21, 2022. [Online]. Available: https://towardsdatascience.com/whats-wrong-with-spectrograms-and-cnns-for-audio-processing-311377d7ccd

[38] Daniel Rothmann, “What’s wrong with CNNs and spectrograms for audio processing?,” Mar. 25, 2018. https://towardsdatascience.com/whats-wrong-with-spectrograms-and-cnns-for-audio-processing-311377d7ccd (accessed Apr. 21, 2022).

[39] S. Kiranyaz, O. Avci, and et al, “1D convolutional neural networks and applications: A survey,” *Mechanical Systems and Signal Processing*, vol. 151, Apr. 2021, doi: 10.1016/j.ymssp.2020.107398.

[40] A. van den Oord *et al.*, “WaveNet: A Generative Model for Raw Audio,” pp. 1–15, 2016, doi: 10.1109/ICASSP.2009.4960364.

[41] Y. Li, † Xiaoshi, and et al, “Text-to-speech Synthesis System based on Wavenet, Stanford University.” 2017.

[42] J. Schmidhuber and F. Gers, “Learning nonregular languages: A comparison of simple recurrent networks and LSTM,” *Neural Computation*, vol. 14, no. 9, pp. 2039–2041.

[43] S. Hochreiter and J. Schmidhuber, “LSTM CAN SOLVE HARD LO G TIME LAG PROBLEMS.”

[44] S. Hochreiter and J. Schmidhuber, “Long Short-term Memory,” *Neural Comput*, vol. 9, pp. 1735–1780, Apr. 1997, doi: 10.1162/neco.1997.9.8.1735.

[45] H. Weerts, A. Mueller, and J. Vanschoren, “Importance of Tuning Hyperparameters of Machine Learning Algorithms,” Jul. 2020, [Online]. Available: http://arxiv.org/abs/2007.07588

[46] N. Lavesson and P. Davidsson, “Quantifying the Impact of Learning Algorithm Parameter Tuning.” [Online]. Available: www.aaai.org

[47] G. Luo, “A review of automatic selection methods for machine learning algorithms and hyper-parameter values,” *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 5, no. 1, Dec. 2016, doi: 10.1007/s13721-016-0125-6.

[48] A. Gulati and J. Qin, “Conformer: Convolution-augmented Transformer for Speech Recognition,” May 2020, [Online]. Available: http://arxiv.org/abs/2005.08100

[49] Y. Lu and Z. Li, “Understanding and improving transformer from a multi-particle dynamic system point of view,” *arXiv preprint arXiv:1906.02762*, 2019.

[50] Z. Wu and Z. Liu, “Lite transformer with long-short range attention,” *arXiv preprint arXiv:2004.11886*, 2020.

[51] Y. Wang, A. Mohamed, and et al., “Transformer based acoustic modeling for hybrid speech recognition,” *arXiv preprint arXiv:1910.09799*, 2019.

[52] G. Synnaeve and Q. Xu, “End-to-End ASR: from Supervised to Semi-Supervised Learning with Modern Architectures,” *arXiv*, vol. 3, 2020.

[53] S. Karita and N. Chen, “A comparative study on transformer vs rnn in speech applications.,” *arXiv preprint arXiv:1909.06317*, 2019.

[54] Q. Zhang and H. Lu, “Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss,” *arXiv*, 2020.

[55] W. Han, Z. Zhang, and et al, “ContextNet: Convolutional Neural Networks for Automatic Speech Recognition with Global Context,” *INTERSPEECH*, 2020.

[56] D. S. Park *et al.*, “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition,” Apr. 2019, doi: 10.21437/Interspeech.2019-2680.

[57] W. Chan, N. Jaitly, and et al, “Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition,” *ICASSP*, 2016.

[58] C. Gulcehre, O. Firat, and et al, “On Using Monolingual Corpora in Neural Machine Translation,” Mar. 2015, [Online]. Available: http://arxiv.org/abs/1503.03535

[59] A. Zeyer, K. Irie, and et al, “Improved training of end-to-end attention models for speech recognition,” in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2018, vol. 2018-September, pp. 7–11. doi: 10.21437/Interspeech.2018-1616.

[60] J. Li and v. Lavrukhin, “Jasper: An End-to-End Convolutional Neural Acoustic Model,” *arXiv:1904.03288*, vol. 3, Aug. 2019.

[61] K. Irie, R. Prabhavalkar, and et al, “On the Choice of Modeling Unit for Sequence-to-Sequence Speech Recognition,” *arXiv:1902.01955*, vol. 2, Jul. 2019.

[62] C.Weng and J. Cui, “Improving Attention Based Sequence-to-Sequence Models for End-to- End English Conversational Speech Recognition,” *NTERSPEECH*, 2018.

[63] A. Zeyer, A. Merboldt, and et al, “A comprehensive analysis on attention models,” *NIPS: Workshop IRASL*, 2018.

[64] D. S. Park *et al.*, “Improved Noisy Student Training for Automatic Speech Recognition,” May 2020, doi: 10.21437/Interspeech.2020-1470.

[65] v. Panayotov, G. Chen, and et al, “Librispeech: An ASR corpus based on public domain audio books,” *ICASSP*, 2015.

[66] J. Kahn, M. Rivi`ere, and et al., “LIBRI-LIGHT: A BENCHMARK FOR ASR WITH LIMITED OR NO SUPERVISION,” *arXiv:1912.07875*, vol. 1, Dec. 2019.

[67] C. Luscher, E. Beck, and et al., “RWTH ASR systems for librispeech: Hybrid vs attention - w/o data augmentation,” *Interspeech*, 2019.

[68] W. Hsu, A. Lee, and et al, “Self supervised speech recognition via local prior matching,” *arXiv*, 2020.

[69] S. Ling, Y. Liu, and et al, “Deep Contextualized Acoustic Representations for Semi-Supervised Speech Recognition,” in *ICASSP 2020*, 2020, pp. 6429–6433. doi: 10.1109/ICASSP40776.2020.9053176.

[70] A. Laptev and R. Korostik, “You Do Not Need More Data: Improving End-To-End Speech Recognition by Text-To-Speech Data Augmentation,” May 2020, doi: 10.1109/CISP-BMEI51763.2020.9263564.

[71] Y. Wang and R. Skerry-Ryan, “Tacotron: Towards end-to-end speech synthesis,” *arXiv preprint arXiv:1703.10135*, 2017.

[72] J. Shen and R. Pang, “Natural tts synthesis by conditioning wavenet on mel spectrogram predictions,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4779–4783.

[73] A. Rosenberg and Y. Zhang, “Speech Recognition with Augmented Synthesized Speech,” in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), SG, Singapore*, 2019, pp. 996–1002.

[74] N. Rossenbach and A. Zeyer, “Generating Synthetic Audio Data for Attention-Based Speech Recognition Systems,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain*, May 2020, pp. 7069–7073.

[75] J. Li and R. Gadde, “Training Neural Speech Recognition Systems with Synthetic Speech Augmentation,” 2018. [Online]. Available: https://nvidia.github.io/OpenSeq2Seq/html/speech-recognition/wave2letter.html

[76] T.-S. Nguyen, S. Stueker, J. Niehues, and A. Waibel, “Improving sequence-to-sequence speech recognition training with on-the-fly data augmentation,” Oct. 2019, [Online]. Available: http://arxiv.org/abs/1910.13296

[77] C. Du and K. Yu, “Speaker Augmentation for Low Resource Speech Recognition,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7719–7723. doi: 10.1109/ICASSP40776.2020.9053139.

[78] J. Shen and R. Pang, “Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions,” in *IEEE ICASSP*, 2018, pp. 4779–4783.

[79] Diederik P. Kingma and Max Welling, “Auto-Encoding Variational Bayes,” *arXiv:1312.6114v10*, May 2014.

[80] Oleksii Kuchaiev, Boris Ginsburg, and et al, “Openseq2seq: extensible toolkit for distributed and mixed precision training of sequence-to-sequence models,” *arXiv:1805.10387*, Nov. 2018, Accessed: Apr. 20, 2022. [Online]. Available: https://arxiv.org/pdf/1805.10387.pdf

[81] Ronan Collobert and Christian Puhrsch, “Wav2Letter: an End-to-End ConvNet-based Speech Recognition System,” *arXiv:1609.03193*, vol. 2, Sep. 2016.

[82] Y. Gao, R. Singh, and B. Raj, “Voice Impersonation using Generative Adversarial Networks,” Feb. 2018, [Online]. Available: http://arxiv.org/abs/1802.06840

[83] A. Polyak, L. Wolf, and Y. Taigman, “TTS Skins: Speaker conversion via ASR,” *arXiv:1904.08983*.

[84] S. Kriman *et al.*, “QuartzNet: Deep Automatic Speech Recognition with 1D Time-Channel Separable Convolutions,” Oct. 2019, [Online]. Available: http://arxiv.org/abs/1910.10261

[85] R. Prenger, R. Valle, and B. Catanzaro, “WaveGlow: A flow-based generative network for speech synthesis,” 2018.

[86] H. Zen, v. Dang, and et al, “LibriTTS: A corpus derived from LibriSpeech for text-to-speech,” *arXiv:1904.02882*, 2019.

[87] J.S. Chung, A. Nagrani, and A. Zisserman, “VoxCeleb2: Deep speaker recognition,” *INTERSPEECH*, 2018.

[88] J. Yamagishi, J. Lorenzo-Trueba, Tomoki Toda, and et al, “Voice Conversion Challenge 2018,” 2018. http://vc-challenge.org/vcc2018/index.html (accessed Apr. 21, 2022).

[89] Lorenzo Trueba, Jaime Yamagishi, and Junichi Toda, “The Voice Conversion Challenge 2018: database.” https://datashare.ed.ac.uk/handle/10283/3061 (accessed Apr. 21, 2022).

[90] J. S. Bradley, “Classroom acoustics for acceptable speech recognition: A review,” *Canadian Acoustics*, vol. 30, no. 3, pp. 42–43, 2002.

[91] M. Benzeghiba, R. de Mori, and et al, “Automatic speech recognition and speech variability: A review,” *Speech Communication*, vol. 49, no. 10–11, pp. 763–786, 2007, doi: 10.1016/j.specom.2007.02.006.

[92] J. Li, L. Deng, Y. G., and H. Reinhold, “An overview of noise-robust automatic speech recognition,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 22, no. 4, pp. 745–777, 2014, doi: 10.1109/TASLP.2014.2304637.

[93] Z. Tuske and P. Golik, “Acoustic Modeling with Deep Neural Networks Using Raw Time Signal for LVCSR,” in *INTERSPEECH*, Sep. 2014, pp. 890–894.

[94] W. Byrne, E. Knodt, and L.D. Consortium, “Hispanic-English Database,” 2014.

[95] Cynthia Clopper, David Pisoni, and L.D Consortium, “Nationwide Speech Project,” 2007.

[96] J. Garofolo, et al, and L. D. Consortium., “TIMIT Acoustic-Phonetic Continuous Speech Corpus,” 1993, [Online]. Available: https://catalog.ldc.upenn.edu/LDC93S1

[97] W. Fisher, G. Doddington, and L.D. Consortium, “NTIMIT,” 1993.

[98] A. Ragni, K. M. Knill, and et al, “Data augmentation for low resource languages,” 2014.

[99] P. C. Loizou, *Speech enhancement: theory and practice*. CRC Press, 2013.

[100] C. Kim and R. M. Stern, “Power-Normalized Cepstral Coefficients (PNCC) for Robust Speech Recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 7, pp. 1315–1329, 2016, doi: 10.1109/TASLP.2016.2545928.

[101] S. Xue, H. Jiang, L. Dai, and Q. Liu, “Speaker Adaptation of Hybrid NN / HMM Model for Speech Recognition Based on Singular Value Decomposition,” *Journal of Signal Processing Systems*, vol. 82, no. 2, pp. 175–185, 2016, doi: 10.1007/s11265-015-1012-6.

[102] S. Sun, B. Zhang, L. Xie, and Y. Zhang, “An unsupervised deep domain adaptation approach for robust speech recognition,” *Neurocomputing*, vol. 257, pp. 79–87, 2017, doi: 10.1016/j.neucom.2016.11.063.

[103] Y. Ning, C. Xing, and L.-J. Zhang, “Domain Knowledge Enhanced Error Correction Service for Intelligent Speech Interaction,” in *Artificial Intelligence and Mobile Services -- AIMS 2019*, Springer International Publishing, 2019, pp. 179–187.

[104] I. Rebai, Y. Benayed, W. Mahdi, and J.-P. Lorre, “Improving speech speech recognition recognition using using data data augmentation augmentation and and acoustic acoustic Improving model fusion fusion model,” *Procedia Computer Science*, vol. 112, pp. 316–322, 2017, doi: 10.1016/j.procs.2017.08.003.

[105] D. Dean and A. Kanagasundaram, “The QUT-NOISE-SRE protocol for the evaluation of noisy speaker recognition,” 2015. Accessed: Apr. 20, 2022. [Online]. Available: https://eprints.qut.edu.au/85240/

[106] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, “A Comparison of Sequence-to-Sequence Models for Speech Recognition,” in *Interspeech*, 2017, pp. 939–943. doi: 10.21437/Interspeech.2017-233.

[107] C. Myers, L. Rabiner, and A. Rosenberg, “Performance Tradeoffs in Dynamic Time Warping Algorithms,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 6, pp. 623–635, 1980.

[108] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Poceedings of theIEEE*, vol. 77, no. 2, pp. 257–286, 1989, doi: 10.1109/5.18626.

[109] T. Liu *et al.*, “Learning to detect a salient object.,” *IEEE Trans Pattern Anal Mach Intell*, vol. 33, no. 2, pp. 353–67, 2011, doi: 10.1109/TPAMI.2010.70.

[110] S. Hochreiter and J. Urgen Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[111] J. V Psutka, “Benefit of Maximum Likelihood Linear Transform (MLLT) Used at Different Levels of Covariance Matrices Clustering in ASR Systems,” in *Text, Speech and Dialogue*, 2007, pp. 431–438.

[112] D. Povey *et al.*, “The Kaldi speech recognition toolkit,” in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011, pp. 1–4. doi: 10.1017/CBO9781107415324.004.

[113] G. Zhong, H. Song, and et al., “The USTC_NELSLIP System for OpenASR21 Challenge.”

[114] H. Song, G. Zhong, R. Wang, C. Wang, J. Du, and L. Dai, “The zxy System for OpenASR21 Challenge.”

[115] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” *The IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016, doi: 10.1109/CVPR.2016.265.

[116] G. Csurka, “Domain Adaptation for Visual Applications: A Comprehensive Survey,” Feb. 2017, [Online]. Available: http://arxiv.org/abs/1702.05374

[117] A. Ramponi and B. Plank, “Neural Unsupervised Domain Adaptation in NLP---A Survey,” May 2020, [Online]. Available: http://arxiv.org/abs/2006.00632

[118] D. Wang and T. F. Zheng, “Transfer Learning for Speech and Language Processing,” Nov. 2015, [Online]. Available: http://arxiv.org/abs/1511.06066

[119] Wei Hu, Zhiyuan Li, and Dingli Yu, “SIMPLE AND EFFECTIVE REGULARIZATION METHODS FOR TRAINING ON NOISILY LABELED DATA WITH GENERALIZATION GUARANTEE,” Apr. 2020. Accessed: Apr. 21, 2022. [Online]. Available: https://iclr.cc/Conferences/2020

[120] W. M. Kouw and M. Loog, “A review of domain adaptation without target labels,” Jan. 2019, doi: 10.1109/TPAMI.2019.2945942.

[121] S. Sun, B. Zhang, L. Xie, and Y. Zhang, “An unsupervised deep domain adaptation approach for robust speech recognition,” *Neurocomputing*, vol. 257, pp. 79–87, Sep. 2017, doi: 10.1016/j.neucom.2016.11.063.

[122] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani, “Training generative neural networks via Maximum Mean Discrepancy optimization,” May 2015, [Online]. Available: http://arxiv.org/abs/1505.03906

[123] H. Phan *et al.*, “THIS LETTER HAS BEEN ACCEPTED FOR PUBLICATION IN IEEE SIGNAL PROCESSING LETTERS 1 Improving GANs for Speech Enhancement.” [Online]. Available: http://github.com/pquochuy/idsegan.

[124] A. Chatziagapi *et al.*, “Data augmentation using GANs for speech emotion recognition,” in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2019, vol. 2019-September, pp. 171–175. doi: 10.21437/Interspeech.2019-2561.

[125] X. Zheng, T. Chalasani, K. Ghosal, S. Lutz, and A. Smolic, “STaDA: Style Transfer as Data Augmentation,” Sep. 2019, doi: 10.5220/0007353401070114.

[126] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-VECTORS: ROBUST DNN EMBEDDINGS FOR SPEAKER RECOGNITION.” [Online]. Available: http://www.openslr.org.

[127] A. van den Oord *et al.*, “WaveNet: A Generative Model for Raw Audio,” pp. 1–15, 2016, doi: 10.1109/ICASSP.2009.4960364.

[128] C. Chadebec and S. Allassonniere, “Data Augmentation with Variational Autoencoders and Manifold Sampling,” *arXiv:2103.13751*, vol. 3, Sep. 2021.

[129] D. Povey *et al.*, “The Kaldi speech recognition toolkit,” *IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 1–4, 2011, doi: 10.1017/CBO9781107415324.004.

[130] Lance Norskog, Guido van Rossum, and et al, “SoX - Sound eXchange.” http://sox.sourceforge.net/ (accessed Apr. 21, 2022).

[131] P. Baldi and K. Hornik, “Neural networks and principal component analysis: Learning from examples without local minima,” *Neural Networks*, vol. 2, no. 1, pp. 53–58, 1989, doi: 10.1016/0893-6080(89)90014-2.

[132] A. L. Maas, Q. V. Le, T. M. O’Neil, O. Vinyals, P. Nguyen, and A. Y. Ng, “Recurrent Neural Networks for Noise Reduction in Robust ASR.,” in *Interspeech*, 2012, pp. 3–6. doi: 10.1016/j.patcog.2005.01.025.

[133] J. Gonzalez and A. Carpi, “Early effects of smoking on the voice: a multidimensional study.,” *Med Sci Monit*, vol. 10, no. 12, pp. CR649-56, 2004, doi: 4738 [pii].

[134] M. Schröder, “Emotional speech synthesis: A review,” in *Source*, 2001, vol. 1, no. 3, pp. 2–5. doi: 10.1002/hyp.6973.

[135] David Snyder, Daniel Gargia-Romero, and et al, “X-VECTORS: ROBUST DNN EMBEDDINGS FOR SPEAKER RECOGNITION,” in *ICASSP 2018*, 2018, pp. 5329–5333.