Semi-Supervised Learning with Generative Models for Pathological Speech Classification

NAM H. TRINH, B.Eng.

Supervised by Dr. DARRAGH O'BRIEN

A thesis submitted in fulfilment of the requirements for the award of Master of Science (M.Sc)

DCU
Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

School of Computing

Dublin City University

June 2021

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Master of Science is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: (Candidate) ID No.: 18215144 Date: 02/06/2021

Acknowledgements

I would like to express my immense gratitude toward my supervisor - Dr. Darragh O'Brien for his thorough guidance during this early stage of my research career and his inspiration to motivate me to further pursue a research career in computer science. I really appreciate his support from the day when I needed documents for a visa application, the day I first arrived at DCU and all of the meticulous edits in my papers and in this thesis. I have learnt so much from his mindset and his guidance about research skills.

I also want to thank the ADAPT Centre for their support during the Covid-19 pandemic through organizing virtual workshops and giving me the opportunity to present my work. I really appreciate the helpful and supportive suggestions from the ADAPT research students and fellows when I was having an opportunity to present my preliminary results. The ADAPT Centre for Digital Content Technology is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund. This work is funded by Science Foundation Ireland under grant No. 17/RC/PHD3488.

I want to take this opportunity to thank my beloved family and friends in Vietnam for their mental support via long but supportive video calls during the difficult Covid-19 pandemic time.

I am thankful to my housemates - Le Tu Khiem, Ninh Van Tu, Nguyen Manh Duy and Nguyen Thao Nhu - for their great support and their great gaming skills. I would like to thank Nguyen Nhu An and Tran Thi Bich Diem for being great neighbours and

sometimes, teaching me how to cook. Finally, I am grateful to all of my Vietnamese friends at DCU for letting me be part of a Vietnamese community and mitigating my homesickness.

Nam H. Trinh
June 2021

Contents

List of Figures

List of Tables

Semi-supervised learning with generative models for pathological speech classification

Nam H. Trinh

Abstract

Recent work in pathological speech classification has employed supervised learning algorithms such as neural networks and support vector machines to classify speech as healthy or pathological. A challenge in applying such machine learning techniques to pathological speech classification is the labelled data shortage problem. While labelled data are expensive and scarce, unlabelled data are inexpensive and plentiful. Labelled data acquisition often entails significant human effort and time-consuming experimental design. Further, for medical applications, privacy and ethical issues must be addressed where patient data is collected.

In this thesis, we investigate a semi-supervised learning (SSL) approach that employs a generative model to incorporate both labelled and unlabelled data into the training process. Generative models explored include both a generative adversarial network (GAN) and a variational autoencoder (VAE). To employ a GAN, we modify its traditional discriminator to not only differentiate between real and fake speech samples but to also classify the given sample as healthy or pathological. To employ a VAE, we first pre-train the VAE with unlabelled data and subsequently, incorporate the pre-trained encoder into a classifier to be trained on labelled data.

We test our approach using three commonly used pathological speech datasets: the Spanish Parkinson's Diseases Dataset (SPDD), the Saarbrucken Voice Database (SVD) and the Arabic Voice Pathology Database (AVPD). We compare the performance of the GAN and VAE-based approaches trained on both labelled and unlabelled data with a traditional supervised approach based on a convolutional neural network (CNN) trained only on labelled data.

We observe that our SSL-based approach leads to an accuracy gain compared to a baseline CNN trained only on labelled pathological speech data. This promising result shows that our approach has the potential to alleviate the labelled data shortage problem in pathological speech classification and other medical applications where labelled data acquisition is challenging.

> Artificial Intelligence is the new electricity.
>
> *Professor Andrew Ng*

Chapter 1

Introduction

In this chapter we present the research questions our work seeks to investigate and describe the structure of the thesis. We also highlight our key contributions and publications. The chapter is organised as follows: in section 1.1 we outline the motivation for our work and set it in context; in section 1.2 we define our research questions; in section 1.3 we describe the structure of the thesis and map our research questions to specific chapters; in section 1.4 we list our contributions; section 1.5 lists publications arising from our work; section 1.6 concludes and summarizes the chapter.

1.1 Motivation and research context

Deep learning for healthcare applications has attracted significant research effort in recent years [11–13]. For example, deep neural networks have been employed in medical imaging (e.g. medical image segmentation and object detection within X-ray images) and achieved state-of-the-art performance in tasks such as Corona virus detection from X-ray images [14, 15], chest pathology classification [16], metal artifact reduction in metal transplants [17] and bone structure segmentation [18]. Besides image-related applications, neural networks for pathological speech classification have also attracted significant research effort recently where supervised machine learning algorithms are applied to classify speech data samples as pathological and healthy [19–22]. Before the advent of deep learning, research in pathological speech classification typically employed a range of traditional machine learning algorithms including Random Forests and Support Vector Machines (SVM) [23–26].

A challenge in the field of machine learning for medical applications is the scarcity of labelled training data [27–30]. Labelled medical data acquisition often demands significant human expertise and can give rise to privacy and ethical concerns. While quality labelled data availability is often limited, unlabelled data are plentiful. Semi-supervised learning (SSL), incorporating both labelled and unlabelled data [10, 31], presents a potential means of alleviating the labelled data shortage problem and thus, improving overall classification performance in pathological speech classification.

Convolutional Neural Networks (CNNs) [1] have been widely applied for computer vision tasks and have achieved considerable success. However, CNNs require large training datasets to perform well. Recently, generative models such as the Generative Adversarial Network (GAN) (introduced in [5]) and the Variational Autoencoder (VAE) (introduced in [32]) have been applied in SSL with benchmark image datasets, e.g. MNIST, CIFAR-10, and SHVN.

In this thesis, we explore both a GAN-based and a VAE-based SSL approach to mitigating the data shortage problem for pathological speech classification. We evaluate the latter proposed approaches by comparing the performance of each against that of a baseline CNN.

1.2 Research questions

Research question 1: *Can deep neural networks outperform traditional machine learning algorithms in pathological speech classification?*

Before the advancement of deep learning, work in pathological speech classification typically employed traditional machine learning algorithms such as Random Forests and Support Vector Machines (SVM) [23–26]. Speech data was first processed to extract salient features. The extracted features were fed into classification algorithms for pathology detections. Thanks to the advances in deep learning, neural networks can now play the role of feature extractor simplifying the speech processing task. To answer our first research question, we extract spectrograms from speech data and feed them (as 2-D matrices) into a CNN. We compare the performance of CNN with that of traditional machine learning algorithms such as SVMs and Random Forests.

Research question 2: *Can we employ generative models (GAN and VAE) to incorporate unlabelled data into the training process and thereby boost pathological speech classification accuracy?*

Generative models were first employed in computer vision to generate highly realistic images for tasks such as human face generation [33, 34], handwritten digit generation [9] and image-to-image translation [7, 35]. Semi-supervised learning approaches employing generative models have also been proposed to alleviate the data shortage problem [36, 37]. While generative models attempt to capture the training data distributions, they also learn the features of these data. To answer our second research question, we incorporate unlabelled data into the training by employing generative models to learn a feature representation from these unlabelled data. We then incorporate these learned features into a supervised neural network trained on labelled data to improve the performance of the supervised model in the scenario where there is a lack of labelled data. The performance of semi-supervised approaches is compared with that of baseline CNNs trained only on labelled data.

Research question 3: *Which generative model (GAN or VAE) yields a better performance boost?*

We compare the performance of the two generative models, the GAN and VAE, in the semi-supervised learning tasks. Consistent training configuration is maintained across all experiments to ensure that experimental results are comparable. We use the librosa framework [38] for speech processing and Keras on top of Tensorflow [39] for neural network model building.

1.3 Thesis structure

The thesis approaches the above research questions as follows:

- Chapter 1 (this chapter) defines our research questions, thesis structure and contributions.
- Chapter 2 presents a literature review on pathological speech classification.
- Chapter 3 presents the deep learning background that forms the basis of our experiments in Chapters 4-6.
- Chapter 4 addresses research question 1.
- Chapter 5 addresses research question 2 in the context of a GAN.
- Chapter 6 addresses research question 2 in the context of a VAE and concludes by addressing research question 3.
- Chapter 7 concludes the thesis, summarising its contributions, limitation and setting our future work.

1.4 Contributions

Our contributions are:

- A baseline deep learning-inspired approach to pathological speech classification employing a CNN trained only on labelled data (we use it as baseline result to assess the performance of the SSL approaches employing a GAN and a VAE),
- a framework for applying GAN-based semi-supervised learning for pathological speech classification,
- a framework for applying VAE-based semi-supervised learning for pathological speech classification,

- empirical experiments comparing the performance of the proposed approaches with that of a baseline CNN using three popular pathological speech datasets.

1.5 Related publications

Several chapters in this thesis are extended versions of the following contributions:

- Chapter 4 has been published in Trinh, N., O'Brien, D. (2019). Pathological speech classification using a convolutional neural network. IMVIP 2019: Irish Machine Vision & Image Processing, Technological University Dublin, Dublin, Ireland, August 28-30. doi:10.21427/9dnc-n002.

- Chapter 5 has been published as Trinh, Nam and O'Brien, Darragh (2020). Semi-supervised learning with generative adversarial networks for pathological speech classification. In Proceedings of the 31st Irish Signals and Systems Conference (ISSC2020), 11 - 12 June 2020, Letterkenny, Ireland (Virtual).

- An extended version of the aforementioned paper is published in Trinh N.H., O'Brien D. (2020) Generative Adversarial Network-Based Semi-supervised Learning for Pathological Speech Classification. In: Espinosa-Anke L., Martín-Vide C., Spasić I. (eds) Statistical Language and Speech Processing. SLSP 2020. Lecture Notes in Computer Science, vol 12379. Springer, Cham.

1.6 Summary

In this chapter we described the context and motivations for our research. We defined the three research questions that this thesis seeks to investigate. We presented the thesis structure and mapped its content to our research questions. We highlighted our contributions and listed related publications.

Somewhere, something incredible is waiting to be known.

Professor Carl Sagan

Chapter 2

Literature Review: Machine Learning for Pathological Speech Classification

In this chapter, we provide a review of speech production and speech pathologies and subsequently, on related work that employs machine learning algorithms for pathological speech classification. We first describe how speech is produced through a three-stage process occurring in our respiratory system. Since the application of deep learning to classify speech pathologies is the focus of our research, a review of speech pathologies and associated voice abnormalities is presented along with some commonly used pathological speech datasets. It is against such datasets that we test our classification methods. This chapter is organized as follows: we review speech production process and speech pathologies in section 2.1; in section 2.2.1 we summarize several commonly used pathological speech datasets; in section 2.2.2 we review pathological speech classification work highlighting datasets, features, classifier design and resulting classification accuracy; Section 2.3 summarizes the chapter.

2.1 Speech production and speech pathologies

2.1.1 Speech production

Speech production transforms thoughts into a speech signal at both the psycholinguistic level, where a cognitive linguistic process in the brain forms words, grammatical structures and meaning, and the motor sensor level, where the articulation initiated at the lungs actually creates actual speech signals that travel to listeners' ears [40]. At the psycholinguistic level, Broca's area in the human brain processes language information including choice of words and grammar structure and sends control signals to motor organs for producing speech [41, 42]. After receiving the signal from the human brain, speech signals are formed by a three-stage process requiring the collaboration of a complicated system of organs including the lungs, trachea, vocal tract, tongue, teeth and nasal cavity. Below, we briefly review the physiology of speech production.

Anatomy of speech production

The three stages of speech production are respiration, phonation and articulation.

During respiration, the lung creates an airflow and pushes it through trachea. Respiration includes inhalation and exhalation. During normal respiration, the durations of inhalation and exhalation are equal. During the speech production process, exhalation dominates [43].

Once the airflow is pushed through the trachea, it enters the larynx where the phonation takes place. During phonation, the larynx modifies the airflow generated by the lungs to produce an acoustic signal by modifying the length of the vocal folds and vibrating the vocal folds to control the airflow through the glottis [43].

During articulation, the vocal tract shapes this airflow to create the desired sound [43, 44]. There are two types of articulatory organs: active articulators and passive articulators. Active articulators include the lower lip, tongue, glottis and uvula that move actively during the speech production process. The active articulators are supported by passive articulators (organs that do not move during speech production) such as the teeth, alveolar ridge, palate and pharynx.

Articulatory phonetics

Articulatory phonetics refers to the way the speech sounds (phonemes) are formed. There are two main types of phoneme: vowels and consonants [45].

Vowel sounds are formed by pushing the airflow from the lungs through vibrating vocal folds [43]. Different resonant frequencies in the vocal tract are created by changing its shape and the position of the tongue and lips. Unlike during consonant formation, there is no restriction in airflow. There are two types of vowel sounds: monophthongs (one vowel, e.g. cat – /cat/ or sit - /sit/) and diphthongs (two vowels e.g. my - /mai/, or brain - /brein/). Three main characteristics describe a vowel sound, namely height, backness and roundedness. Height and backness refers to tongue position. For example, /i/ is produced with a high tongue position whereas /a/ is produced with a low tongue position. Producing /i/ requires the tongue to be forward while producing /u/ requires the tongue to be backward. Roundedness refers to the shape of the lips when the sound is produced, e.g. /u/ is a rounded vowel since the lips are rounded when pronouncing /u/ while /i/ is an unrounded vowel.

Consonants are formed also by forcing an airflow from the lungs but, unlike with vowels, the vocal tract is now constricted [43]. Different consonant sounds are created through changes in place and manner of articulation and voicing. Place of articulation refers to the position at which the airflow is constricted: bilabial consonants such as /p/, /b/ and /m/ are formed by touching the two lips whereas labiodental consonant sounds such as /f/ and /v/ are produced by touching the upper teeth with the lower lip. The manner of articulation refers to the way in which the airflow constriction occurs, e.g. stop consonant sounds /p/, t/ and /k/, fricatives /θ/, /v/ and /z/, affricative sounds (a combination of a stop and fricative) /tʃ/ and nasals (airflow through the nasal cavity) /m/ and /n/. Voicing refers to the vibration of the vocal folds, i.e. if the vocal folds vibrate, the sound is voiced (/b/ or /z/) and if the vocal folds do not vibrate, the sound is unvoiced (/p/ or /s/).

In the experiments described in this thesis, we make use of sustained vowel sounds for pathological speech classification. Patients' speech was collected by requiring them to pronounce vowel sounds (/a/, /e/ or /i/) and sustain this sound for several seconds.

2.1.2 Speech pathologies

Speech production is a complex process at both cognitive and motor levels and speech pathologies may arise at either level. In this section, we provide a description of several speech pathologies including speech disorders caused by Reinke's edema, vocal cord paralysis, nodules, sulcus, dysarthria and Parkinson's Disease.

Reinke's edema

Reinke's edema (RE) is the swelling of the vocal cords caused by edema (a type of fluid) that occupies the Reinke's space [46]. "Reinke's Edema" is named after the anatomist Reinke who conducted morphological studies investigating the condition. Patients with RE often have an abnormally low speaking fundamental frequency and high subglottal pressure [47]. One major symptom of RE is a hoarseness and deepening of the voice [48]. The patient's voice has an abnormally low fundamental frequency due to an increase in the edema within the Reinke's space, leading to a lower frequency vibration of the vocal cords. A study in [46] has shown that tobacco use is the primary risk factor for RE.

Vocal cord paralysis

Vocal cord paralysis occurs when the nerve system (recurrent laryngeal nerves) that controls the muscles of the larynx malfunctions, leading to paralysis of the vocal cords [49]. Vocal cord paralysis affects not only the ability to produce speech but also the ability to breathe since the vocal cords also protect our respiratory system by preventing food and drink from entering the trachea. Noticeable symptoms of vocal cord paralysis include difficulties in breathing, swallowing and speaking [50, 51]. Such symptoms arise from the difficulty in moving the muscles in the larynx and the vocal cords. Causes of vocal cord paralysis include congenital issues (neurological disorders at birth), viral infection, tumors, trauma, thyroid diseases and other neurological disorders [51, 52].

Vocal cord nodules

Vocal cord nodules refer to a speech pathology involving the abnormal growth of masses on the vocal cords that affect their vibration during the speech production process [53, 54]. The nodules are benign, i.e. not cancerous. The quality of voice is affected by nodules as they increase the mass of the vocal cords, affecting their vibration and closure. In patients with vocal cord nodules, the fundamental frequency of their voice may be affected and the range of pitch frequency may be reduced. Symptoms of vocal cord nodules include hoarseness, breathiness and a reduced range of voice [53].

Sulcus vocalis

Sulcus vocalis results from a crease or groove on the vocal cords caused by a reduction in the thickness (or sometimes even the absence) of a special layer of tissue on the vocal cords called the superficial lamina propria [55]. According to clinicians, sulcus vocalis is classified into three types: Type I with moderate loss at lamina propria; Type II (sulcus vergeture) and Type III (sulcus vocalis) with severe destruction of lamina propria causing severe dysphonia due to a completely lack of vibration of the vocal cord [56]. Symptoms of sulcus vocalis include hoarseness and breathiness of voice.

Dysarthria

Dysarthria is a speech disorder caused by neuromuscular deterioration [57]. Speech production involves the movement of muscles in the face, lips, tongue and throat. When dysarthria occurs, the control signal transmitted from the brain to these muscles weaken [58, 59]. This may be due to brain damage at birth or after an illness or injury [58]. Symptoms of dysarthria include abnormal speed of speech (too fast or too slow), difficulty in moving the tongue, lips and jaw, mumbled or choppy speech, and hard-to-understand speech.

Parkinson's Disease

Parkinson's Disease (PD) is a neurodegenerative disease caused by neuronal loss in a region of the brain called the substantia nigra [60]. The loss of neurons responsible for producing dopamine causes a dopamine deficit that leads to abnormal brain activity.

Several causes of the latter neuronal loss have been put forward including environmental factors and genetics; however, none have been conclusively proven. In terms of environmental factor, a meta-analysis in [61] claimed that smoking, caffeine and alcohol consumption may reduce the risk of PD while exposure to pesticide and other environmental toxins may increase the risk of PD. Several genetic studies have proposed that gene mutation, notably in the α-synuclein and leucine-rich repeat kinase two genes, may be associated with PD [62].

Symptoms of PD include tremor, bradykinesia (slow movement), muscular rigidity and postural instability [63]. Speech and voice disorders are also regularly observed in PD patients. A study of 200 Parkinson's suffers found that 90% had a dysfunction in their vocal tracts and larynx when producing speech [64]. A longitudinal study in [65] proposed that a decrease in pitch variability can be observed as an early symptom of PD. A study in [66] showed that speech disfluencies in PD patients were significantly greater than in healthy controls in a reading task.

Scope of the thesis

Our focus in this thesis is on the application of deep learning techniques for the early detection of voice pathologies such as those described above. Early diagnosis of such conditions invariably leads to improved patient outcomes and lower overall healthcare costs. We make use of publicly available datasets that contain samples of the above pathologies. Those datasets and previous approaches to pathological speech classification are described below.

2.2 Machine learning for pathological speech classification

The process of pathological speech classification is illustrated in Figure 2.1. The first step is salient feature extraction from raw speech. During feature extraction, raw speech signals are typically converted from the time-domain into frequency-domain features (by means of, for example, the Fourier Transform). Frequency-domain features are subsequently supplied to a classifier that differentiates between healthy and pathological speech. Below we review publicly available datasets and previous machine learning-based approaches to pathology classification.

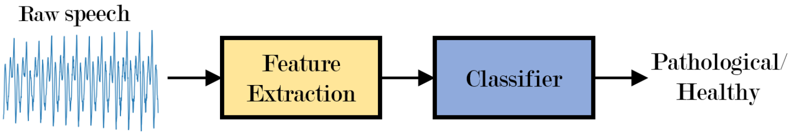


Figure 2.1: A general pathological speech classification system

2.2.1 Pathological speech datasets

We summarize in Table 2.1 the contents of several commonly used speech pathology datasets. We compare their contents in terms of languages, constituent phonemes, size and classes (healthy versus pathological).

Table 2.1: Commonly used pathological speech datasets

Datasets	Language	Phonemes	Size	Classes
Saarbrucken Voice Database (SVD) [67]	German	Sustained vowels /a/, /i/, /u/ and conversational sentences	Over 2000 speech samples (33% of them are healthy samples and 67% of them are pathological samples)	Two classes: healthy samples and samples of different pathologies.
Massachusetts Ear and Eye Infirmary (MEEI) [68]	English	Sustained vowel /a/ and Rainbow passage	Total 1400 speech recordings of sustained vowel sounds /a/ (from 653 pathological speakers and 53 healthy controls)	Two classes: healthy samples and pathological samples.
Arabic Voice Pathology Database (AVPD) [69, 70]	Arabic	Three vowels /a/, /i/, /u/, isolated words and conversational speech	Total 366 samples of normal and pathological speech (51% of them are normal, the rest are pathological)	Two classes: healthy samples and pathological samples.
Child Pathological and Emotional Speech Database (CPESD) [71]	French	Conversational sentences	2542 speech samples collected from 99 children (of which 12 with AD, 13 with SLI, 10 with PDD-NOS, and the rest are TD children	Four classes: Typically Developing (TD), Autism Disorder (AD), Pervasive Developmental Disorder - Not Otherwise Specified (PDD-NOS), Specific Language Impairment (SLI).
Spanish Parkinson's Disease Dataset (SPDD) [72]	Spanish	Sustained vowels, isolated words and conversational speech	50 people with Parkinson's Disease and 50 healthy control	Two classes: healthy and Parkinson's Disease speech samples.
VOice ICar fEDerico II Database (VOICED) [73, 74]	Italian	Sustained vowel /a/	208 samples (consisting of 150 pathological speech samples and 58 healthy control samples)	Two classes: healthy samples and pathological samples.

2.2.2 Pathological speech classification

In Table 2.2 we summarize related work in machine learning-based speech pathology classification. For each study we highlight datasets used, features extracted, classifier design and reported classification accuracy.

Before the era of deep learning, early work in the area of pathological speech classification typically required the design and implementation of analysis methods in order to acquire meaningful features for classification.

In [23], the authors extracted features such as pitch, jitter, shimmer and harmonic-to-noise ratio before feeding them into a linear regression model to discriminate between PD and healthy speech samples.

In [24], the authors extracted 132 features including jitter, shimmer, recurrence period density entropy, noise features and mel-frequency cepstral coefficients (MFCCs) that were subsequently fed into classification algorithms such as random forests (RFs) and support vector machines (SVMs). They reported a classification accuracy of 93.0% with RF and an accuracy of 98.6% with SVMs.

In [25], the authors proposed a classification system in which jitter, shimmer, glottal pulse parameters, pitch, amplitude parameters and harmonicity parameters (autocorrelation, harmonic-to-noise ratio) are extracted and input to SVM and K-Nearest Neighbors.

In [75], a method was proposed to analyse Parkinson's Disease based on fundamental and formant frequencies (F0, F1, F2, F3), jitter, shimmer, harmonic-to-noise ratio, soft phonation index and glottal features. Using an SVM for classification, an accuracy of 93% was achieved with this method.

In [76], a feature extraction method was proposed to extract harmonic-to-noise ratio contour on sentence-level speech data; the classification accuracy achieved with this method was 87.3% using an SVM classifier.

With the emergence of deep learning algorithms, pathological speech classification models based on neural networks have also been proposed. For example, in [77–80], MFCCs serve as input vectors to a Multi Layer Perceptron (MLP).

MLP drawbacks, however, include overfitting and a potentially long training period due to the large number of model parameters. To address MLP-related issues, Convolutional Neural Network (CNN)-based models were proposed. Using a CNN-based approach (with a CaffeNet architecture), a state-of-the-art result of 97.5% accuracy with the Saarbrucken Voice Database was reported consisting of 1616 pathological speech samples and 686 normal speech samples in the form of sustained vowel /a/ [81]. In our previous work in [82], we achieved promising results with 99.0% accuracy on SVD and with 96.7% accuracy on SPDD. In experiments described in the Chapters 5 and 6, we employ the VGG-16-based CNN model proposed in [81] as a baseline model.

Besides SVD and SPDD, pathological speech classification methods have been applied with other datasets. In [83,84], various acoustic features were proposed and their ability to detect speech disorders in children was evaluated against the CPESD (see Table 2.1). In [85], a voice disorder system for mobile devices was trained on MEEI and VOICED (see Table 2.1) and achieved a promising classification accuracy.

In this thesis, we employ three pathological speech datasets for all experiments: SVD, SVDD and AVPD since such datasets are publicly available and commonly used for pathological speech classification.

2.3 Summary

In this chapter, we reviewed the speech production. We described the speech pathologies normally covered by publicly available datasets including those arising from Reinke's edema, vocal cord paralysis, nodules, sulcus vocalis, dysarthria and Parkinson's Disease. We presented related work in pathological speech classification by firstly summarizing the contents of commonly used pathological speech datasets and secondly, reviewing machine learning techniques employed for classifying pathological from healthy speech. Work to date in the area of pathological speech classification has largely as-

Table 2.2: Related work in pathological speech classification: features, classifiers and reported accuracy

Reference	Dataset	Features	Classifier	Accuracy
Poorjam et al. (2018) [26]	Data collected by the authors in collaboration with Sage Bionetworks	MFCCs	SVM	88.0%
Moon et al. (2018) [77]	SVD [67]	Jitter, shimmer and MFCCs	MLP	87.4%
Smitha et al. (2018) [78]	Supplied by the Nitte Institute of Speech and Hearing Mangaluru	MFCCs	MLP	95.0%
Shia et al. (2017) [80]	SVD	Wavelet Sub-band Energy Coefficients	MLP	93.3%
Alhussein et al. (2018) [81]	SVD	Spectrogram (after framing and applying STFT)	CNN	97.5%
Trinh et al. (2019) [82]	SVD SPDD [70]	Spectrogram	CNN	99.0% 96.7%

sumed that an adequate corpus of training samples (including both normal and pathological speech) is available to the model to be trained. In this thesis we shift focus to exploring how a lack of training data can be mitigated through semi-supervised learning.

Our intelligence is what makes us human, and AI is an extension of that quality.

Professor Yann LeCun

Chapter 3

Background: Deep Neural Networks, Generative Adversarial Networks and Variational Autoencoders

In this chapter, we present the background material on deep learning including neural networks and generative models that underpins the research reported in subsequent chapters. The chapter is organized as follows: in section 3.1 we provide a review of neural networks including multi-layer perceptrons; in section 3.2 we present an overview of convolutional neural networks; in sections 3.3 and 3.4 we review the architecture and cost functions of deep generative models such as generative adversarial networks and variational autoencoders and provide a literature review of related work employing generative models; in section 3.5 we review commonly used semi-supervised learning approach; section 3.6 summarizes and concludes the chapter.

3.1 Deep neural networks

Deep neural networks (DNNs) or Multi-layer perceptrons (MLPs) are fundamental models in deep learning. In a supervised learning scheme, the goal of training an MLP is to approximate a function f to map from an input x to an output y such that $y = f(x)$ and subsequently, to apply this function f to unseen input x to predict an output [86]. The main building blocks of an MLP are neurons placed across different layers: input layer, hidden layers and output layer (as depicted in Figure 3.1). The goal is to train the network to obtain optimal parameters θ such that the network yields outputs close to the ground truths (or labels).

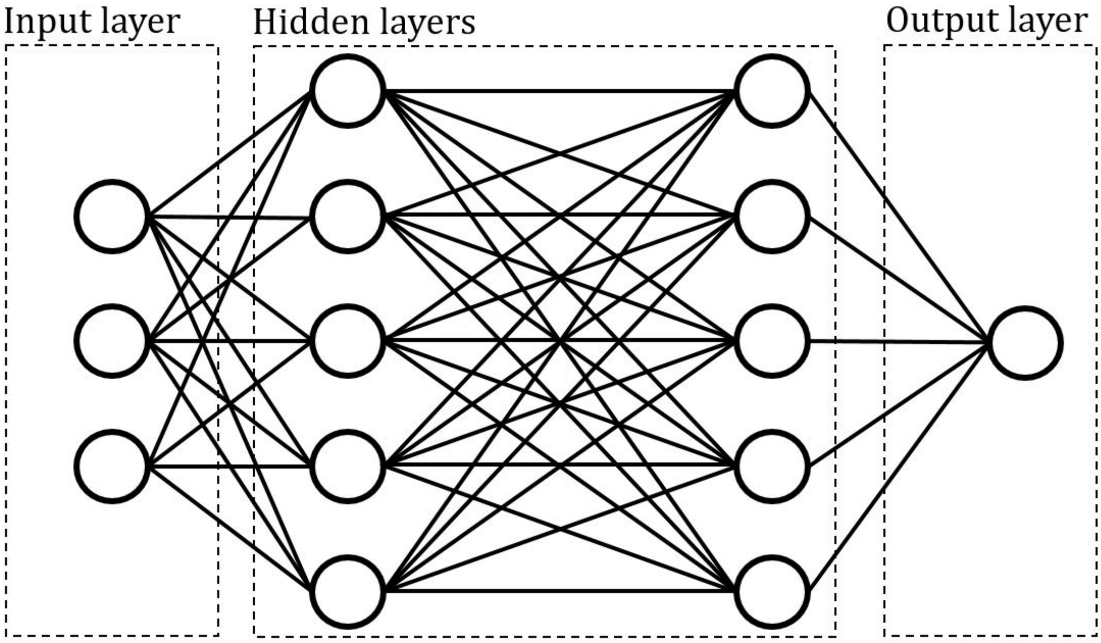


Figure 3.1: MLP architecture

3.1.1 Feed-forward propagation

In feed-forward propagation, a neuron takes input vector $\mathbf{x}$, multiplies $\mathbf{x}$ by weight vector $\mathbf{W}$ and adds the result to the bias b (the result is $\mathbf{z} = \mathbf{W^T}\mathbf{x} + b$) [87]. The weighted sum z is then acted upon by an activation function. The output of a neuron is mathematically expressed as follows:

$$y = g(\mathbf{W^T x} + b) \tag{3.1}$$

Feed-forward propagation is so-called since the data x flows into the input layer, through hidden layers and exits at the output layer. Some commonly used activation functions are softmax, sigmoid, tanh, Rectified Linear Unit (ReLU), Leaky Rectified Linear Unit (LeakyReLU). We briefly describe the properties of each in section 3.1.3.

3.1.2 Loss functions

The goal of feed-forward propagation during the training phase is to output a predicted value $\hat{y}$ that is as close as possible to the actual output y. To evaluate the closeness, we implement a loss function. Several commonly used loss functions include the mean absolute error (MAE), the mean squared error (MSE), the cross entropy (CE) and the Kullback-Leibler divergence.

Mean absolute error (MAE) is the average absolute difference between predicted outputs and actual outputs over a training set and is defined as follows:

$$MAE(y, \hat{y}) = \frac{1}{m}\sum_{i=1}^{m} ||\hat{y}_{(i)} - y_{(i)}|| \tag{3.2}$$

where $y_{(i)}$ is the actual output for the i^{th} sample, $\hat{y}_{(i)}$ is the predicted output, and m is the number of training samples.

Mean squared error (MSE) is the average squared difference between predicted outputs and actual outputs over a training set and is defined as follows:

$$MSE(y, \hat{y}) = \frac{1}{m}\sum_{i=1}^{m} (\hat{y}_{(i)} - y_{(i)})^2 \tag{3.3}$$

where $y_{(i)}$ is the actual output of the i^{th} sample, $\hat{y}_{(i)}$ is the predicted output, m is the number of training samples.

Compared to MAE, MSE is larger for outlier data points since the difference between the predicted and actual output is squared. Thus, MSE is highly sensitive to outliers while MAE is less affected by their presence [88].

Cross-entropy (CE) measures the performance of a model where the model's output is the probability of an event from 0 to 1 and is defined as follows:

$$CE(y, \hat{y}) = -\frac{1}{m}\sum_{i=1}^{m}[y\log(\hat{y}) - (1-y)\log(1-\hat{y})] \tag{3.4}$$

Kullback-Leibler (KL) divergence is the distance between two probability distributions [89]. A KL divergence between two discrete distributions P and Q is defined as follows:

$$D_{KL}(P||Q) = \sum_{i} P(i)\log\frac{P(i)}{Q(i)} \tag{3.5}$$

3.1.3 Activation functions

Neural network training employs a gradient-based optimization method (described in more detail in section 3.1.4) to update parameters θ by computing the derivative of the loss function with respect to each parameter. The design of activation functions plays an important role as the derivative is computed by propagating backward through the network. In this section we briefly describe some commonly used activation functions for neural networks and their properties.

Sigmoid activation

A sigmoid activation is expressed as follows:

$$g(z) = \sigma(z) = \frac{1}{1+e^{-z}} \tag{3.6}$$

The range of the sigmoid activation is $[0, 1]$ which can be interpreted as the probability of a binary variable. The sigmoid saturates as the input z grows more positive or negative. The function is fully differentiable. The disadvantages of the sigmoid activation include a vanishing gradient when employing gradient-based optimization and being computationally expensive. Since training a neural network is a gradient-based optimization (see section 3.1.4), derivatives must be computed for the backpropagation process. However, the sigmoid activation exhibit slow convergence due to its widespread saturation, i.e. it saturates to 0 with negative inputs and saturates to 1 with positive inputs, making the derivative at these saturation regions close to 0.

Hyperbolic tangent

A hyperbolic tangent or *tanh* activation is expressed as follows:

$$g(z) = tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{3.7}$$

Similar to sigmoid, the *tanh* activation saturates to -1 with increasingly negative input z and saturates to 1 with increasingly positive input z. The main advantage of tanh over sigmoid is its similarity to the identity function, i.e $tanh(0) = 0$ while $\sigma(0) = \frac{1}{2}$. Since it resembles the identity function, training a network with *tanh* is as straightforward as training a network with linear activation if the input value z is small.

Rectified linear unit

A rectified linear unit (ReLU) activation [90] is a piecewise linear function that outputs 0 if $z < 0$ and outputs z if $z > 0$ where z is the activation input. ReLU is mathematically expressed as follows:

$$g(z) = max(0, z) \tag{3.8}$$

Its linearity makes ReLU straightforward to optimize since it is linear when the input z is greater than 0 and is also computationally efficient. However, the "dying ReLU" problem arises where the input z approaches 0 or becomes negative, causing the derivative of the function to move close to 0 and preventing convergence through gradient descent [91]. To mitigate this problem, several revised ReLU versions have been proposed including one that adds a negative slope when the input z is less than

0 [92]. In [93], empirical experiments have shown that adding a negative slope to a standard ReLU can improve the performance of a convolutional neural network on an image classification task. A LeakyReLU [94] is another extended ReLU activation and is expressed as follows:

$$g(z) = max(\alpha z, z) \tag{3.9}$$

where α is a small fixed value (the default value for α is 0.01). Another ReLU version is the Parametric ReLU or PRELU [92] and it too is expressed as:

$$g(z) = max(\alpha z, z) \tag{3.10}$$

where α however is now a trainable parameter, i.e. the slope of the negative PRELU component is trainable.

Softmax activation

Applying the softmax activation to a vector of M elements normalizes them into a probability distribution with M probabilities [88]. Unlike other activation functions applied for a single neuron, a softmax activation is typically applied to a layer of neurons. Softmax is implemented as follows:

$$g(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{3.11}$$

3.1.4 Gradient-based backpropagation

After initializing the feed-forward propagation model including its weight and bias, backpropagation is applied to train the network, i.e. to update its parameters until the loss function is minimized using optimization algorithms [95]. Optimization algorithms minimize or maximize an objective or cost function $f(\theta)$ by iteratively adjusting θ until the cost function reaches its minimum. In the context of deep neural networks, we employ optimization algorithms to iteratively update the trainable parameters θ (weight matrix $\mathbf{W}$ and bias vector b) of the networks such that the loss function J is minimized. The loss function is a function of trainable parameters of the network:

$$J(W, b) = \frac{1}{m} \sum_{i=1}^{m} L(y_{(i)}, \hat{y}_{(i)}) \tag{3.12}$$

where L is the loss function applied for each training sample, $\hat{y}$, y are the predicted and actual output of the i^{th} sample, respectively, and m is the number of training samples. We briefly review below some commonly used optimization algorithms for neural networks.

Gradient descent is a first-order iterative optimization method to find the minimum of the loss function by updating the parameters $\mathbf{W}$ and b [95]. To find a local minimum of a function $f(x)$ through gradient descent, we calculate the derivative $f'(x)$ and then update x:

$$x := x - \alpha f'(x) \tag{3.13}$$

where α is a pre-defined constant. In deep neural networks, $f(x)$ corresponds to the loss function $J(\mathbf{W}, b)$ and α is the learning rate. The gradient descent algorithm for neural networks is defined in Algorithm 1.

Algorithm 1: Gradient Descent

```
Initialize parameters $\mathbf{W}, b$
for number of epochs do
    • Feed-forward propagation to compute predicted output $\hat{y}$
    • Calculate the loss function $J(\mathbf{W}, b)$
    • Compute the gradients of the loss function with respect to
      network parameters, i.e. $d\mathbf{W}$ and $db$
    • Update parameters
      $\mathbf{W} := \mathbf{W} - \alpha dW$
      $\mathbf{b} := \mathbf{b} - \alpha db$
```

where $dW = \dfrac{\partial J(\mathbf{W}, b)}{\partial \mathbf{W}}$ and $db = \dfrac{\partial J(\mathbf{W}, b)}{\partial b}$ are partial derivatives of the loss function $J(\mathbf{W}, b)$ with respect to weights $\mathbf{W}$ and bias b.

Adaptive momentum optimization (Adam) [96] is an extension of mini-batch gradient descent employing a combination of two techniques: momentum and root mean square (RMS) prop. A momentum technique employs a moving average function to calculate the average partial derivative over a batch [88, 97]. A moving average function calculates the average over n successive values:

$$A[t] = \beta A[t-1] + (1-\beta)X[t] \tag{3.14}$$

where $A[t]$ is the average value at data point t^{th} and $X[t]$ is the value at that point. This approach saves on the memory needed to store the values to calculate the average value since we only need to store the value $A[t]$ at the time t. In gradient descent with momentum, the moving average calculates the average value of parameter updates after backpropagation, i.e. $\dfrac{\partial J(\mathbf{W}, b)}{\partial \mathbf{W}}$ and $\dfrac{\partial J(\mathbf{W}, b)}{\partial \mathbf{b}}$ as follows:

$$v_{dW} = \beta_1 v_{dW} + (1-\beta_1) d\mathbf{W} \tag{3.15}$$

$$v_{db} = \beta_1 v_{dW} + (1-\beta_1) db \tag{3.16}$$

$$\mathbf{W} := \mathbf{W} - \alpha v_{dW} \tag{3.17}$$

$$\mathbf{b} := \mathbf{b} - \alpha v_{db} \tag{3.18}$$

where $dW = \dfrac{\partial J(\mathbf{W}, b)}{\partial \mathbf{W}}$ and $db = \dfrac{\partial J(\mathbf{W}, b)}{\partial \mathbf{b}}$.

RMS prop calculates the average value of $d\mathbf{W}^2$ and db^2 and divides the updated amounts $d\mathbf{W}$ and db by these average values [98]. The motivation behind this method is to reduce the number of updates, thereby dampening oscillations, when converging on the minimum. RMS prop is mathematically defined as:

$$S_{dW} = \beta_2 S_{dW} + (1 - \beta_2) d\mathbf{W}^2 \tag{3.19}$$

$$S_{db} = \beta_2 S_{dW} + (1 - \beta_2) db^2 \tag{3.20}$$

$$\mathbf{W} := \mathbf{W} - \alpha \frac{dW}{\sqrt{S_{dW}} + \epsilon} \tag{3.21}$$

$$\mathbf{b} := \mathbf{b} - \alpha \frac{db}{\sqrt{S_{db}} + \epsilon} \tag{3.22}$$

where ϵ is a small constant added to the denominator to avoid zero division. Combining momentum and RMS prop, Adam optimization is described as in Algorithm 2. Adam is a popular optimization algorithm due to its fast convergence. In our experiments presented in Chapters 4, 5, and 6, we employ the Adam optimizer with $\alpha = 0.00002$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ as recommended from the original work introducing Adam [96].

3.2 Convolutional neural networks

A convolutional neural network (CNN) [99] is a type of neural network that employs convolutional operations to extract features. Convolutional and pooling layers compute intermediate features from input data before feeding them into fully-connected layers for classification. In this section, we review the operation of a convolutional neural network including convolution and pooling. We also review several classic CNN architectures: LeNet, AlexNet, VGG and ResNet. We finish the section with an overview of related work, mostly in the field of computer vision, employing convolutional neural networks.

Algorithm 2: Adam: Gradient descent with momentum and RMS prop

```
Initialize parameters $\mathbf{W}, b$
Initialize $v_{dW} = 0$ and $v_{db} = 0$
for number of epochs do
    for number of batches do
        • Sample a batch of $m$ samples: $x_{(1)}, x_{(2)}, \ldots, x_{(m)}$
        • Feed-forward propagation to calculate predicted output $\hat{y}$
        • Calculate the loss function $J(\mathbf{W}, b)$
        • Compute the gradients of the loss function with respect to
          network parameters, i.e. $d\mathbf{W}$ and $db$
        • Update parameters
          $v_{dW} = \beta_1 v_{dW} + (1 - \beta_1) d\mathbf{W}$; $v_{dW}^{corrected} = \frac{v_{dW}}{1 - \beta_1^i}$
          $v_{db} = \beta_1 v_{db} + (1 - \beta_1) db$; $v_{db}^{corrected} = \frac{v_{db}}{1 - \beta_1^i}$
          $S_{dW} = \beta_2 S_{dW} + (1 - \beta_2) d\mathbf{W}^2$; $S_{dW}^{corrected} = \frac{S_{dW}}{1 - \beta_2^i}$
          $S_{db} = \beta_2 S_{db} + (1 - \beta_2) db$; $S_{db}^{corrected} = \frac{S_{db}}{1 - \beta_2^i}$
          $\mathbf{W} := \mathbf{W} - \alpha \frac{v_{dW}^{corrected}}{\sqrt{S_{dW}^{corrected}} + \epsilon}$
          $\mathbf{b} := \mathbf{b} - \alpha \frac{v_{db}^{corrected}}{\sqrt{S_{db}^{corrected}} + \epsilon}$
```

3.2.1 CNN operation

Convolution

Convolutional layers often appear early in a CNN. Operating on the input image (or any two-dimensional input data) a convolutional layer is considered as a feature extractor. A convolution operation takes input x and multiplies it with a kernel (or filter) W using the sliding dot product or correlation (as illustrated in Figure 3.2). The results are fed into non-linear activation functions (sigmoid, tanh or ReLU) to incorporate non-linearity in the network. The output of the convolution operation is called a feature map. In CNNs, the kernel in Figure 3.2 is modelled by trainable parameters $\mathbf{W} = [W_1, W_2, \ldots W_k]$ and can be updated via backpropagation. A kernel has typical size of 1×1, 3×3 and 5×5. The convolution operation is mathematically expressed as follows [88]:

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n) \tag{3.23}$$

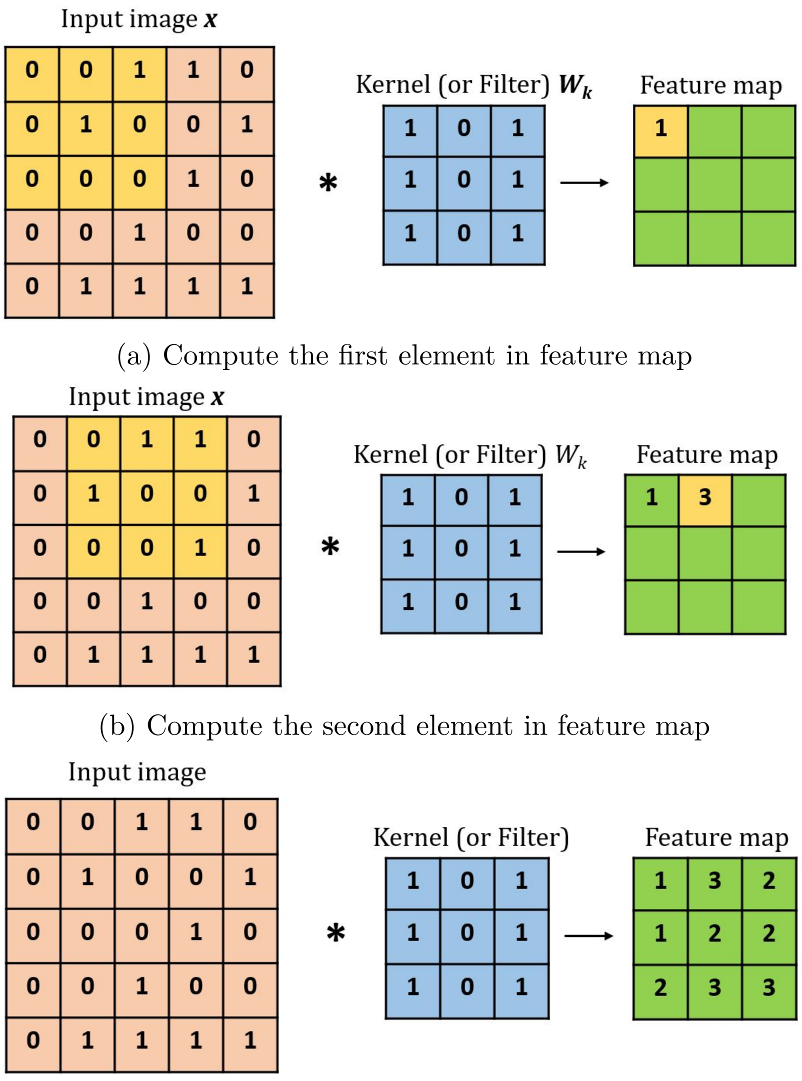
where (i, j) is the position of the currently computed element in the resulting output S, (m, n) is the size of the kernel K and I is the input data.

Compared with fully-connected feed-forward MLPs, CNNs have several advantages including sparsity of connections and parameter sharing. With fully-connected feed-forward MLPs, every neuron in layer N connects with every neuron in the layer $N + 1$, resulting in a large number of connections and consequently model parameters. For example, if the input image has size $32 \times 32 \times 3$ which is then flattened into a vector of $3,072$ input pixels, and if the first hidden layer consists of $1,000$ neurons, then the first layer will have $3,072,000$ trainable weights. In a CNN, the same kernel is reused for multiple regions in the input matrix. Regardless of the size of the input, the number of parameters (the number of trainable weights) in the kernel is constant. This parameter sharing also reduces the density of the connections between layers compared with fully-connected feed-forward MLPs, making CNNs less prone to overfitting.

Pooling operation

Following the convolutional layers, pooling layers are employed to reduce the dimensionality of the feature map. Dimensionality reduction lowers the computational complexity and condenses extracted features by eliminating noise.

Two types of pooling are max pooling and average pooling. Max pooling outputs the maximum values over pooled regions in the feature map (as illustrated in Figure 3.3a). Average pooling outputs the average values over pooled regions in the feature map (as illustrated in Figure 3.3b). In this thesis, we use max pooling in all of our

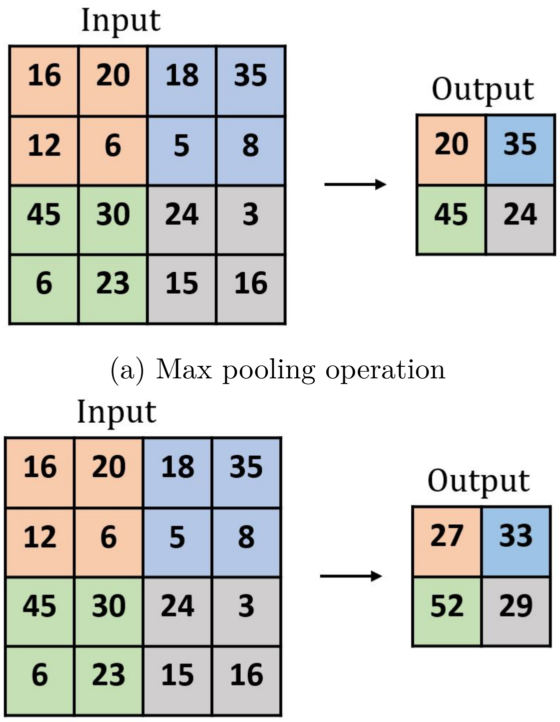


(a) Compute the first element in feature map

(b) Compute the second element in feature map

(c) Complete computations for the feature map

Figure 3.2: Convolution (from [1])



Input

16	20	18	35
12	6	5	8
45	30	24	3
6	23	15	16

Output

20	35
45	24

(a) Max pooling operation

Input

16	20	18	35
12	6	5	8
45	30	24	3
6	23	15	16

Output

27	33
52	29

(b) Average Pooling

Figure 3.3: Pooling (from [1])

CNN designs to reduce the dimensions of the feature map and hence, reduce the computational cost.

3.2.2 Classic convolutional neural networks

CNNs were first introduced in the 1990s for pattern recognition in 2D images. In 2012, AlexNet using convolutional operation and max pooling achieved a state-of-the-art performance with the ImageNet dataset. After AlexNet, several alternative CNN architectures have been developed and evaluated against benchmark datasets such as ResNet, Inception and VGG. In this section, we review the architectures of several classic CNNs including LeNet-5, AlexNet and VGG.

LeNet-5

The Lenet-5 network [2] (as shown in Figure 3.4) consists of seven layers with three convolution layers followed by two pooling layers. The feature map following the third

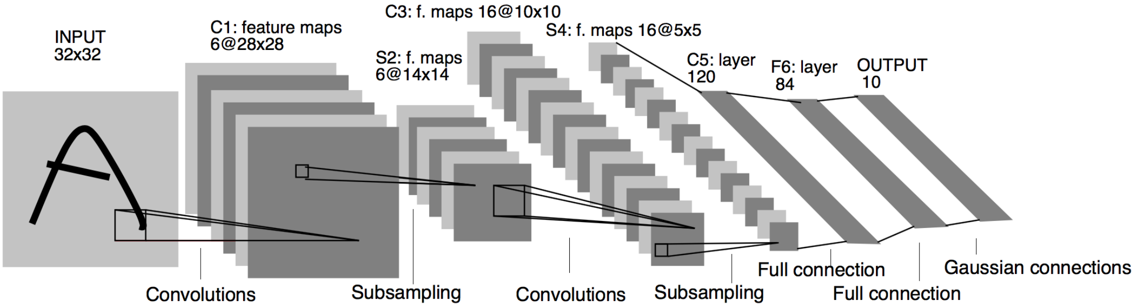


Figure 3.4: LeNet-5 architecture (from [2])

convolution layer is flattened and fed into a fully-connected layer of 120 neurons, followed by an 84-neuron fully-connected layer and a 10-neuron output layer. Lenet-5 was shown to outperform other recognition methods in handwritten character recognition.

AlexNet

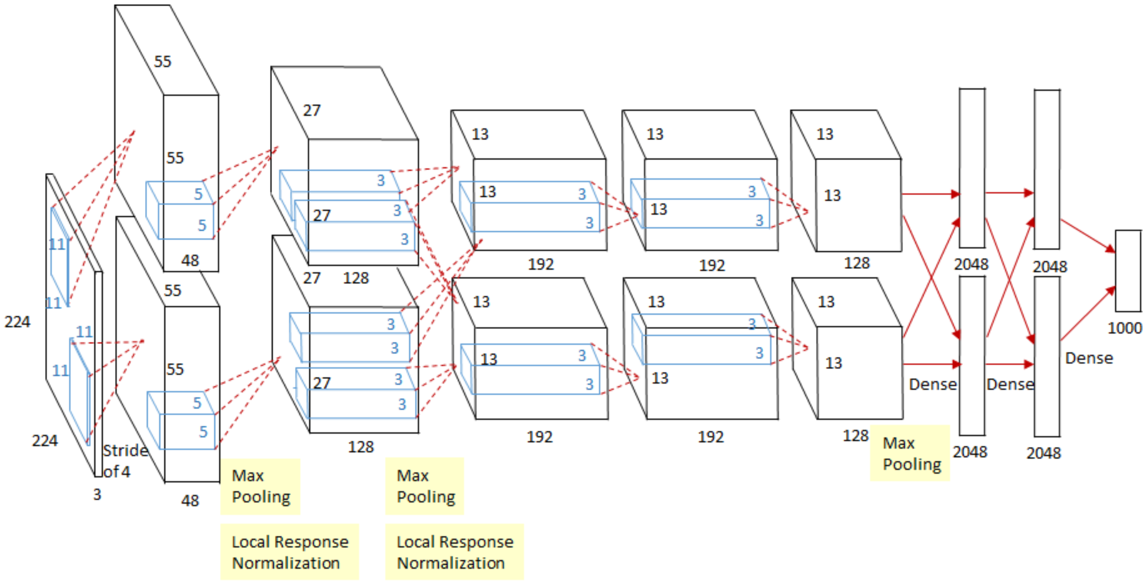


Figure 3.5: AlexNet architecture (from [3])

AlexNet (introduced in [3]) consists of seven layers (as shown in Figure 3.5). The first convolutional layer takes an input image of size 224 × 224 × 3 and applies 96 11 × 11 × 3 filters. The second convolutional layer has 256 filters of size 5 × 5 × 48 followed by a max pooling layer. The third, fourth and fifth convolutional layers have respectively 384, 384 and 256 filters of size 3 × 3 × 192. After the fifth convolutional layer, a max pooling layer reduces the dimensionality of data before feeding it into two fully-connected 4096-neuron layers. The final output layer has 1,000 neurons to output the predicted probability for 1,000 classes in the ImageNet dataset. AlexNet won first place by achieving a state-of-the-art performance (15.3% top-5 error) in the 2012 ImageNet competition.

VGG

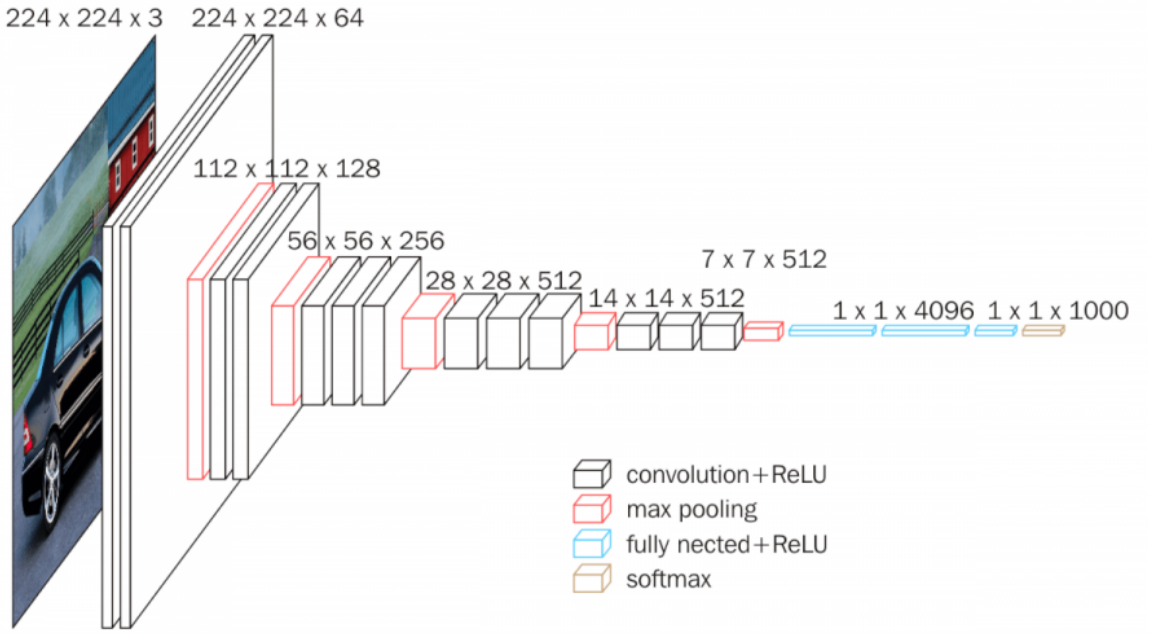


Figure 3.6: VGG-16 architecture (from [4])

The VGG CNN (introduced in [4]) consists solely of 3 × 3 filters instead of using all filter sizes as in the Inception network or large filter sizes as in AlexNet. The depth of the network ranges from 16-19 layers. In this thesis, we take the VGG-16 based network (its architecture is shown in Figure 3.6) as one of our baseline model against which we compare the performance of the proposed approaches.

3.3 Generative adversarial networks

3.3.1 Overview

The original GAN [5] architecture is illustrated in Figure 3.7. A GAN is a generative model taking random noise as input and seeking to generate as output samples from a real data distribution. A vanilla GAN consists of a discriminator and a generator. The generator takes random noise as input and generates new data samples. The discriminator's objective is to discriminate between real and generated samples (provided by the generator), classifying them as real or fake, respectively. The two networks compete until an equilibrium is reached where the discriminator cannot reliably discriminate between real and fake data.

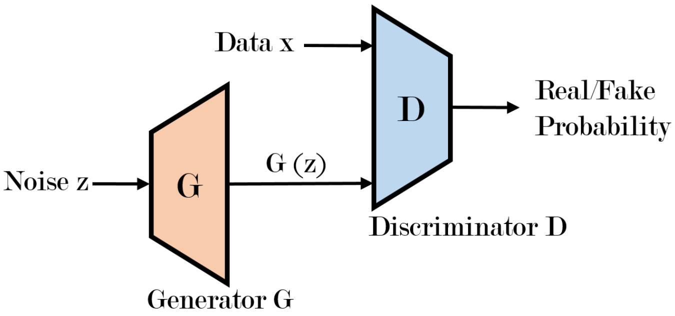


Figure 3.7: GAN architecture (from [5])

3.3.2 Loss function

Let D be the discriminator and G be the generator. The minimax game between D and G is modelled mathematically as follows:

$$\min_G \max_D V(G,D) = E_{x \sim p_{data}(x)}[logD(x)] + E_{z \sim p_z(z)}[log(1 - D(G(z)))] \tag{3.24}$$

where $E_{x \sim p_{data}(x)}$ is the expected value over all real data samples with a data distribution $p_{data}(x)$, $D(x)$ is the probability that a real data sample is classified as real,

$E_{z \sim p_z(z)}$ is the expected value over all noise samples with a prior noise distribution $p_z(z)$, $G(z)$ is the generated output from input noise z.

A GAN's training process is described in Algorithm 3. The objective of the training process is to train D to maximize the probability of classifying generated samples $G(z)$ as fake and data samples x as real and to train G to convince D that generated samples, $G(z)$, are real. In other words, D is trained to maximize the loss function (3.24) while G is trained to minimize (3.24).

Algorithm 3: GAN's training process

```
for number of epochs do
    • Sample a batch of $m$ training examples from real data distribution $p_{data}(x)$:
      $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$
    • Sample a batch of $m$ noise samples from noise distribution $p_z(z)$:
      $z^{(1)}, z^{(2)}, \ldots, z^{(m)}$
    • Train the discriminator to minimize the loss function:
          $Loss(D) = -(\mathbb{E}_{x \sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))])$    (3.25)
    • Sample a batch of $m$ noise samples from noise distribution $p_z(z)$:
      $z^{(1)}, z^{(2)}, \ldots, z^{(m)}$
    • Train the generator to minimize the loss function:
          $Loss(G) = \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))]$    (3.26)
```

3.3.3 Related work

Since their introduction in [5], GANs have attracted significant research interest with the typical goals of a) generating highly realistic images and b) learning a feature representation in unsupervised learning for classification tasks. In this section, we review several major milestones in GAN design including improved techniques for training GANs; along with the Info GAN, Cycle GAN, and Style GAN architectures.

Improved techniques for training GANs

Though GANs have the ability to generate highly realistic images compared with other generative models, GANs training process is challenging due to the problems of non-convergence and mode collapse. In [37], new techniques were introduced to improve the stability of GAN training including feature matching, minibatch discrimination, historical averaging, one-sided label smoothing and batch normalization. Incorporating these techniques, a GAN achieved a state-of-the-art performance in semi-supervised learning tasks with benchmark datasets (MNIST and CIFAR10) [37]. The same work also proposed the Inception score – a new evaluation metric for GANs - and provided a basis to evaluate the quality of GAN-generated images.

Deep convolutional generative adversarial networks (DCGAN)

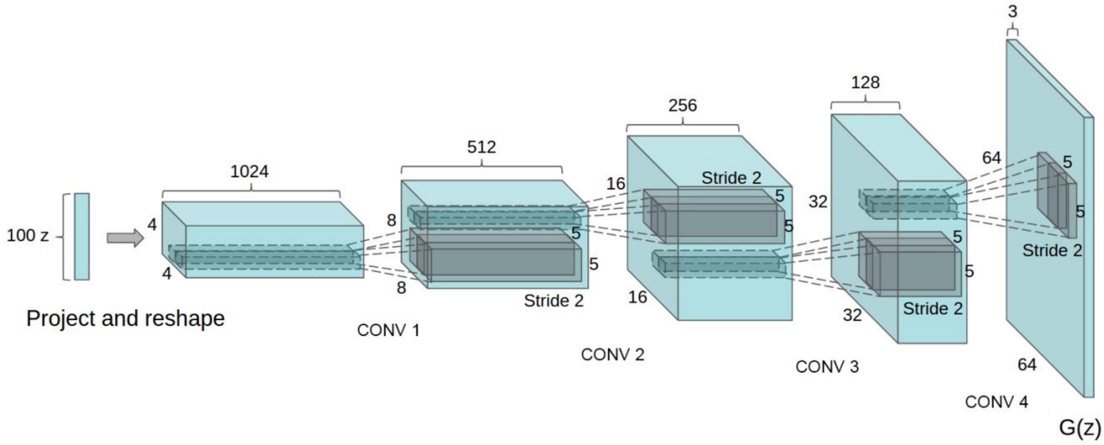


Figure 3.8: DCGAN's generator (from [6])

In [6], Radford et al. proposed a new architecture for GANs (called DCGAN) where both the discriminator and the generator consisted only of convolutional layers instead of multi-layer perceptrons. In DCGAN's generator, all activations are ReLU except for *tanh* at the output layer and no fully-connected layers are used (as illustrated in Figure 3.8). In the discriminator, LeakyReLU replaced ReLU. With this new architecture, DCGAN achieved state-of-the-art classification performance with the SHVN and CIFAR-10 datasets in which a discriminator provided a representation learned from unsupervised pre-training. In this thesis, our GAN design (as described in Chapter 5) is inspired by DCGAN.

Info GAN

In [100], Info GAN was proposed for learning a disentangled representation using information theory where the Info GAN was trained to maximize the mutual information between the latent codes and the observation at the output of the generator. In unsupervised learning, one of the main approaches is to learn from unlabelled data and subsequently, use pre-trained networks for supervised learning tasks. A challenge with this approach is that features learnt from unlabelled data may not be useful for downstream classification tasks. The goal of Info GAN was to learn disentangled representation features from unlabelled data that can be useful for supervised classification tasks. The Info GAN loss function included a regularized term to maximize the mutual information between latent code c and the generator's output $G(z, c)$. The loss function of Info GAN was revised from that of the original GANs as follows:

$$\begin{aligned}\min_G \max_D V(G, D) = E_{x \sim p_{data}(x)}[logD(x)] + E_{z \sim p_z(z)}[log(1 - D(G(z)))] \\ - \lambda I(c; G(z, c))\end{aligned} \tag{3.27}$$

Info GAN successfully manipulated latent features in datasets such as 3D Chairs, CelebA and SHVN and generated new images based on those manipulated features.

Cycle GAN

Cycle GAN [7] is a proposed method to translate images from a domain X to a domain Y (as illustrated in Figure 3.9) without the need for paired images in the training dataset. The goal of Cycle GAN's training process is to learn a mapping from domain X to domain Y and vice versa. Let G be the mapping function from domain X to domain Y and F be the mapping function from domain Y to domain X. Cycle

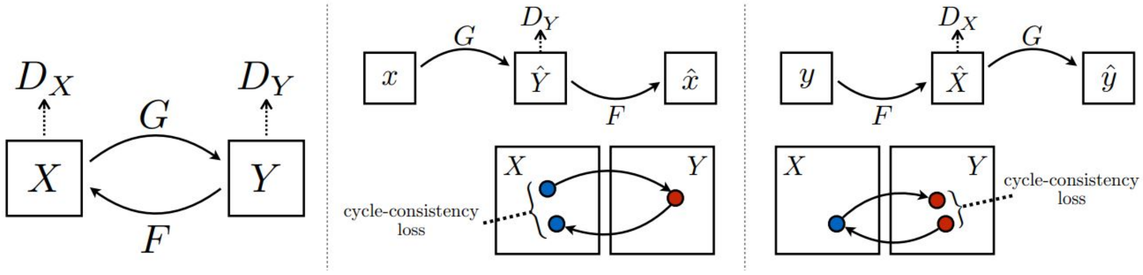


Figure 3.9: Cycle GAN with cycle consistency loss (from [7])

consistency loss was incorporated with adversarial loss to ensure that when translating an image $x \in X$ to domain Y to produce a generated image $G(x)$ and subsequently, translating back the image $G(x)$ to domain X to get another generated image $F(G(x))$ one can retrieve the same image x, i.e. to reproduce an image $F(G(x))$ as close as possible to the initial image x.

Other GAN architectures

In a semi-supervised GAN (SGAN) [101], the discriminator is extended from a binary classifier to a (K+1)-class classifier where K is the number of classes for semi-supervised tasks. In [33], Style GAN was proposed with the latent vector z modified to map the training data distribution before being fed into the generator instead of the Gaussian distribution used in a vanilla GAN. Self-Attention GAN (SAGAN) [102] employed the self-attention mechanism from Natural Language Processing for image generation tasks and achieved a state-of-the-art performance in terms of Inception score with the ImageNet dataset.

3.4 Variational autoencoders

In this section we review the architecture of a standard autoencoder and a variational autoencoder. We also provide a brief literature review of related work employing variational autoencoders as generative models.

3.4.1 Autoencoders

Autoencoders [103] are dimensionality reduction models. An autoencoder consists of an encoder E that attempts to reduce the dimensions of the input data x, encoding x as a latent variable z and a decoder D that attempts to reconstruct the original data x from the latent variable z (as illustrated in Figure 3.10). The dimension of the latent space z is often significantly less than that of the original data x. The goal of the autoencoder's training is to reconstruct the original data x with minimal difference between x and x', i.e. to minimize the reconstruction loss function:

$$L(x, x') = \sum_{m}(x - x')^2 \tag{3.28}$$

where m is the number of input samples

Autoencoders have been widely applied in dimensionality reduction [104, 105], representation learning for unsupervised learning [106–108] and image denoising [109].

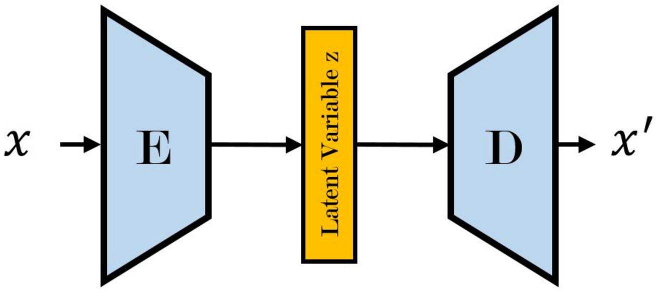


Figure 3.10: Autoencoder

3.4.2 Variational autoencoders

Limitations of standard autoencoders

A major limitation of standard autoencoders is that the latent space at the output of the encoder may not be continuous and if the decoder decodes a vector from an unsmooth region, then the newly generated data may not be realistic. The *variational* autoencoder (VAE) [9, 110] is an extension of the autoencoder whereby variational inference is incorporated as an optimization problem to address the discontinuity in the latent variable distribution. In a VAE, the latent variable z becomes a vector of distributions instead of a vector of fixed values. Similar to a traditional autoencoder, the training of a VAE also minimizes the reconstruction loss between the original input data x and the output of the decoder x'. The difference between a VAE and a standard autoencoder is that the encoder encodes the input data x into distributions rather than as fixed values and subsequently, the decoder decodes a sampled point from the encoded distributions to generate the new data x'.

Probabilistic model of a VAE

The probabilistic model of a VAE is presented in Figure 3.11 where a latent variable z is sampled to generate new data X. The main objective of VAE training is to sample the latent variable z that can produce new data points in X and subsequently, compute $P(X|z)$ from z to generate new data. Given a prior distribution $P(z)$, the posterior

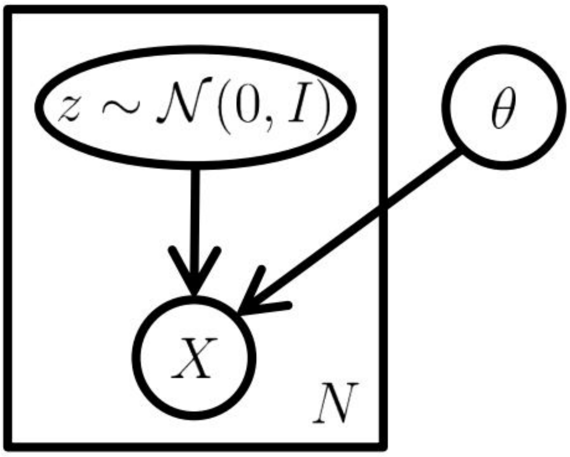


Figure 3.11: A graphical model of a VAE (from [8])

$P(z|X)$ is computed by Bayes' theorem:

$$P(z|X) = \frac{P(X, z)}{P(X)} \tag{3.29}$$

In equation 3.29, the denominator $P(X)$ is intractable since $P(x) = \int_z P(X|z)P(z)dz$. Therefore, variational inference is employed to compute $P(z|X)$ by defining a new distribution $Q(z|X)$ and optimizing $Q(z|X)$ to be as close as possible to the optimal posterior distribution $P(z|X)$. KL divergence is used to evaluate the distance between $Q(z|X)$ and $P(z|X)$. The training goal is to minimize this KL divergence:

$$\begin{aligned} D_{KL}[Q(z|X)||P(z|X)] &= \mathbb{E}_{z\sim Q}[\log Q(z|X) - \log P(z|X)] && (3.30) \\ &= \mathbb{E}_{z\sim Q}[\log Q(z|X) - \log \frac{P(X|z)P(z)}{P(X)}] && (3.31) \\ &= \mathbb{E}_{z\sim Q}[\log Q(z|X) - \log P(X|z) - \log P(z)] + \log P(X) && (3.32) \\ &= -\mathbb{E}_{z\sim Q}[\log P(X|z)] + D_{KL}[Q(z|X)||P(z)] + \log P(X) && (3.33) \end{aligned}$$

Rearranging terms in Equation 3.33 yields:

$$\log P(X) - D_{KL}[Q(z|X)||P(z|X)] = \mathbb{E}_{z\sim Q}[\log P(X|z)] - D_{KL}[Q(z|X)||P(z)] \quad (3.34)$$

The right hand side of Equation 3.34 is called Evidence Lower Bound (ELBO) since due to Jensen's inequality, any KL divergence is greater than 0 [87] and therefore, $\log P(X) \geq ELBO$. The goal of VAE training is to minimize $D_{KL}[Q(z|X)||P(z|X)]$, i.e. to maximize the ELBO.

Architectural overview

The architecture of a VAE is shown in Figure 3.12. Compared to the architecture of a standard autoencoder, the VAE encodes the input data into latent distribution of $z = \mu + \epsilon\sigma$ instead of fixed vectors. Maximizing the ELBO in Equation 3.34 is equivalent to minimizing the loss function:

$$L_{VAE} = -\mathbb{E}_{z\sim Q}[\log P(X|z)] + D_{KL}[Q(z|X)||P(z)] \quad (3.35)$$

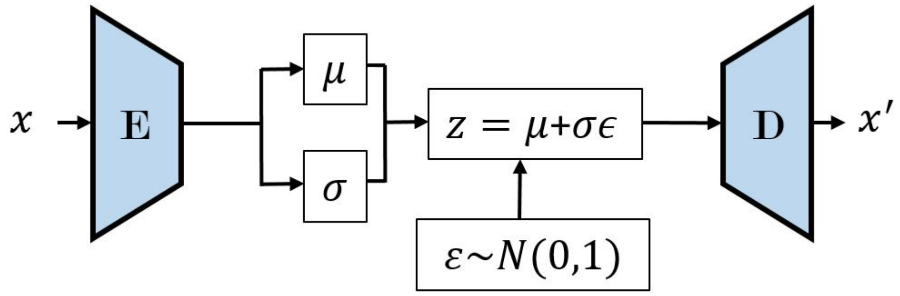


Figure 3.12: Architecture of a VAE (from [9])

In Equation 3.35, the first term is a reconstruction loss to ensure that the decoded output x' matches the input x and the second term is a regularization term where the model is optimized to minimize the divergence between the latent distribution z and the univariate Gaussian distribution ϵ.

3.4.3 Related work

Since the introduction of VAEs in 2013 [9], many applications have employed a VAE as a deep generative model for data generation and for representation learning.

In [111], a deep feature consistent VAE (DFC-VAE) was proposed with the traditional pixel-by-pixel reconstruction loss replaced by feature perceptual loss where the feature representations of input and output images were extracted from a pre-trained deep CNN and the difference was computed with these feature representations. With this improvement, DFC-VAE generated high-quality face images and also achieved a state-of-the-art result in predicting facial attributes.

In [112], grammar VAE (GVAE) was proposed and achieved state-of-the-art performance in generating discrete data such as arithmetic expressions and molecules.

In [113], multimodal VAE (MVAE) was proposed for fake news detection in which the MVAE consists of three main components: an encoder, a decoder and a fake news detector. MVAE achieved state-of-the-art performance with two real world datasets.

In [114], collaborative VAE (CVAE), a Bayesian-based generative model, was proposed to employ both content and rating in a recommender system. Experiments have shown that CVAE achieves state-of-the-art performance compared to other recommendation approaches.

3.5 Semi-supervised learning

Semi-supervised learning is a combination of supervised learning and unsupervised learning incorporating both labelled and unlabelled data into the training process with the goal of improving overall classification performance [10,31]. In this section, we provide a review of traditional semi-supervised learning and semi-supervised learning with generative models (VAEs and GANs). In supervised learning, a dataset consists of labelled data points (x_i, y_i) where x_i are input data points belonging to the input space χ_l and y_i are the labels corresponding to the input. The goal of supervised training is to find a mapping function from data points x_i to corresponding labels y_i. However, in many classification tasks, the quantity of labelled data points is often limited, giving rise to the problem of overfitting as the model fits only to the limited amount of training data and may not perform well on unseen data. Unsupervised learning refers to methods to cluster unlabelled data points $x_i \in \chi_u$ (data points without corresponding labels) into groups that share similar features [115]. Semi-supervised learning combines both types of learning: supervised and unsupervised learning and incorporates both unlabelled and labelled data into the training process to improve the classification performance [10, 31]. We briefly describe three important assumptions in semi-supervised learning below.

3.5.1 Assumptions in semi-supervised learning

Three important assumptions in semi-supervised learning (SSL) are the smoothness assumption, the low density assumption and the manifold assumption [116]. The smoothness assumption relates to the fact that two data points x and x' residing close to each other in the input space X should have the same corresponding label y and

y'. The low density assumption states that the decision boundary of a classification model separating input data points in the input space should not cross over high density regions. The manifold assumption in SSL states that the input space consists of many low-dimensional *manifolds* (low-dimensional representation of data points) and if data points have the same manifolds, they should have the same labels.

3.5.2 Semi-supervised learning methods

In [117], entropy minimization was proposed as an SSL method to ensures that the model (after being trained on labelled data) gives a confident prediction on unlabelled data, i.e. to have a minimum entropy toward predictions of the model on unlabelled data. Intuitively, this method ensures that the supervised model learns useful information and it can *understand* the clustering information of the unlabelled data.

Another popular SSL method is self-training with pseudo-labelling in which a model is first trained on labelled data and subsequently employed to generate predictions on unlabelled data [118]. The most confident generated predictions will be incorporated as *new labelled* data along with the initial labelled data in the next supervised training iteration. Recently the Noisy Student method employed pseudo-labelling by training a "teacher" model on labelled data and subsequently generating new labels for unlabelled data [119]. This method achieved state-of-the-art top-1 accuracy in the ImageNet classification task.

Label consistency using data augmentation refers to the assumption that two or more data points being augmented from the same data point (regardless of being labelled or unlabelled) should have the same label. Several techniques employing label consistency have been proposed. In [120], pi-model and temporal ensembling method were proposed in which a reconstruction loss based on the mean squared error between two augmented data points was incorporated into the overall loss function. Temporal ensembling computes the exponential moving average of label predictions after each epoch; however, if the dataset is large, the moving average of the target becomes computationally expensive. The Mean Teacher method was proposed in [121] to solve this issue by computing the moving average weights of the model instead of label predictions. The Mean Teacher outperformed Temporal Ensembling and achieved a state-of-the-art classification performance with the SHVN and CIFAR-10 datasets. Virtual Adversarial Training (VAT) (proposed in [122]) also employs label consistency by applying adversarial training [123] for unlabelled data to suit the task of semi-supervised learning. VAT achieved state-of-the-art performance in semi-supervised learning tasks with SHVN and CIFAR-10.

Unsupervised processing methods extract features from unlabelled data and, subsequently, incorporate extracted features in supervised models trained on labelled data. A common feature extraction approach is to apply an autoencoder to first train it to reconstruct the original data and, subsequently, incorporate the pre-trained encoder into the supervised model trained on labelled data [124, 125].

Another popular method in SSL is the use of generative models such as generative adversarial networks and variational autoencoders. In the next section, we briefly review related work that employs generative models for the task of semi-supervised learning.

3.5.3 Semi-supervised learning with GANs and VAEs

Generative adversarial networks (GANs) [5] have been employed in SSL and have been shown capable of contributing considerable improvements to overall classification performance using benchmark image datasets such as MNIST, CIFAR-10 and SHVN. In [37], several new architectural features and training procedures were proposed in order to boost GAN performance in a semi-supervised setting. In [126], SSL incorporating a GAN, specifically a Categorical GAN or CatGAN, was proposed. For the SSL task the GAN's discriminator, a binary classifier, was replaced with a $(K + 1)$-class classifier (where K is the number of classes to be classified). This approach demonstrated a significant improvement in accuracy compared to traditional classifiers in image classification tasks. In [101], the proposed GAN-based approach outperformed traditional classifiers at the MNIST classification task. In [127], the proposed GAN-based SSL method with manifold invariance achieved accuracy gains with CIFAR-10 and SHVN datasets. In [128], the proposed GAN method along with a complementary generator improved the overall performance in image classification tasks. Recently, MarginGAN [129] (based on margin theory) achieved high accuracy compared to other SSL methods.

Besides GANs, variational inference generative methods such as variational autoencoders (VAE) have also been tested in an SSL context [36]. In [107], the proposed approach using sequence to sequence autoencoders for representation learning achieved a promising accuracy gain in an acoustic scene classification task.

Semi-supervised approaches have been applied in medical imaging. In [28], the authors report a significant improvement in medical imaging segmentation thanks to SSL. In [29], a graph-based SSL approach incorporating a CNN was proposed for breast cancer diagnosis. In [30], an attention-based SSL approach achieves state-of-the-art results on real clinical segmentation datasets.

Work to-date in pathological speech classification has typically assumed an adequate corpus of pathological speech data. In [130], an approach using GANs was employed to learn feature representations for classifying speech of children with autism spectrum conditions. This approach achieved better results than traditional classification algorithms including linear SVM and MLP. In our previous work [131], we presented preliminary results where a semi-supervised method was applied to mitigate the data shortage problem. In Chapter 5, we further explore and extend the GAN-based SSL approach by testing against three popular pathological speech datasets. In Chapter 6, we employ the VAE-based approach for representation learning and test it against three datasets.

3.6 Summary

In this chapter, we presented background material on neural networks and deep generative models (including GANs and VAEs). This background material provides the context for our experiments in Chapters 4-6. We reviewed the fundamentals of neural networks including multi-layer perceptrons, loss functions, activation functions, the convolution operation and several classic CNN architectures. For GANs, we reviewed their architecture, loss function and several achievements in GAN-related work. For VAEs, we first reviewed standard autoencoders and followed this with a description of VAEs. In the next three chapters, we apply these ideas to the problem of pathological speech classification.

The future depends on some graduate student who is deeply suspicious of everything I have said.

Professor Geoffrey Hinton

Chapter 4

Experiment #1: Convolutional Neural Networks for Pathological Speech Classification

Convolutional neural networks (CNNs) have enabled significant improvements across a number of applications in computer vision such as object detection, face recognition and image classification. An audio signal can be visually represented as a spectrogram that captures the time-varying frequency content of the signal. In this chapter, we seek to answer our first research question: *Can deep neural networks outperform traditional machine learning algorithms in pathological speech classification?* We describe how a CNN, taking as input the spectrogram of an audio signal can learn to distinguish pathological from healthy speech. We propose a CNN structure and implement it using Keras on top of Tensorflow. We test the approach across three commonly used pathological speech datasets. We then compare the performance of our CNN with that of traditional machine learning algorithms including Random Forests and Support Vector Machines. We observe that our CNN outperforms these traditional discriminative algorithms in terms of classification accuracy. The content of this chapter elaborates on the results presented in Trinh, N., O'Brien, D. (2019). Pathological speech classification using a convolutional neural network. IMVIP 2019: Irish Machine Vision & Image Processing, Technological University Dublin, Dublin, Ireland, August 28-30. doi:10.21427/9dnc-n002. The chapter is organised as follows: section 4.1 describes the methodology of using a CNN for pathological speech classification; section 4.2 presents the experiments testing the CNN against three pathological speech datasets; section 4.3 summarises and concludes the chapter.

4.1 Methodology

In this section, we describe the design and implementation of a CNN-based approach to pathological speech classification. We subsequently describe our baseline classification algorithms based on Random Forests and Support Vector Machines.

4.1.1 CNN approach to pathological speech classification

Given in Figure 4.1 are example spectrograms extracted from pathological and healthy speech samples (in this case sustained vowels). Distortion across the pathological speech sample is observed. By contrast, the frequency content of spectrograms from healthy speech samples is more stable. Our goal is to test whether a CNN can detect the distortions and instabilities in the spectrogram indicative of pathological speech.

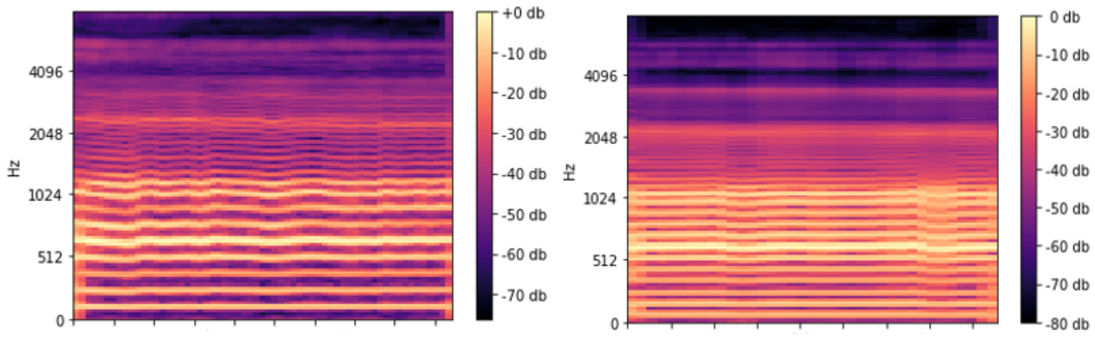


Figure 4.1: Spectrograms of a pathological speech sample (left) and of a healthy speech sample (right)

In this thesis, all spectrograms are calculated using Librosa [38]. The speech signals are first windowed (with a window length of $\approx$ 5.8 ms) and the Short-time Fourier Transform (STFT) is subsequently applied to extract the frequency components of the audio signal. The resulting image is fed to a CNN for classification.

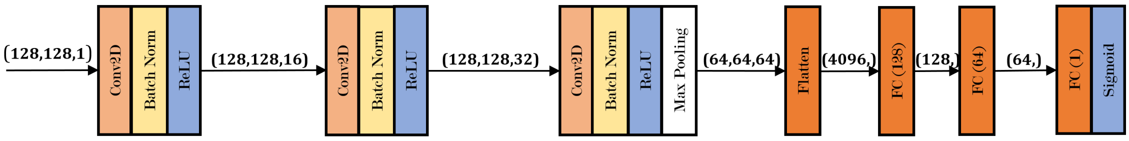


Figure 4.2: The proposed CNN architecture

4.1.2 Baseline classification algorithms

Random forest

A random dorest (RF) is a classification algorithm consisting of a number of decision trees and employing bagging and feature randomness [132]. A decision tree is a tree-structured supervised learning algorithm in which the input data are analysed through a set of if-else conditions at each node of the tree and split into sub-nodes until a leaf node (i.e. a terminal node) is reached [133]. The objective in a random forest is to construct a set of uncorrelated decision trees and subsequently, treat the decisions of each as committee members' votes. Below we briefly describe two core concepts in an RF algorithm: bagging and feature randomness.

- **Bagging** (or bootstrap aggregating) is a statistical sampling technique in which given a sample size n, a set of B bootstrap samples of size n is created by sampling with replacement, e.g. if a sample has data points $(1, 2, 3, 4, 5)$, then a random bootstrap sample may be $(1, 2, 2, 4, 5)$ or $(1, 3, 3, 5, 5)$ [134]. Since decision trees are sensitive to the training data, different training data results in different decision trees. The goal of employing bagging in an RF model is to build *uncorrelated* decision trees as the trees are trained on different bootstrap samples.

- **Feature randomness** In a normal standalone decision tree, when splitting a node, every possible features are considered and the one that best separates the data is chosen. However, in an RF model, each tree selects only a random subset of possible features, resulting in a low correlation between trees in the RF.

Support vector machine

A support vector machine (SVM) is a classification algorithm that aims to find an optimal plane to separate classes of data [135]. The *margin* is the closest distance between the threshold and data points. The objective of an SVM model is to find an optimal hyperplane in N dimensions where N is the number of features with maximized margin. SVMs employ hinge loss as their loss function which is given as:

$$L_{SVM}(x, y) = C\sum_{i}^{N}\max(0, 1 - y_i f(x_i)) \tag{4.1}$$

where C is the regularization parameter, N is the number of samples, x is the input data and y is the data label.

Below we briefly describe three important hyperparameters in an SVM design:

- **Kernel** refers to the method SVMs employ to transform the input data points to a high dimensional feature space [87]. There are four commonly used kernel functions for SVMs: linear kernel, polynomial kernel, radial basis function (RBF) and sigmoid. While a linear kernel is a linear transformation method, the three latter methods are non-linear transformation methods [136].

- **Regularization term** C determines the size of the margin of SVMs. A large value of C adds more penalty to the loss function if a data point is misclassified, leading to a smaller margin. Similarly, a small value of C results in a large margin.

- **Gamma** γ determines the range of points considered to compute the margin. A large gamma leads to the margin being computed from only data points close to the decision boundary and conversely, a small gamma value results in the margin being computed from points far away from the decision boundary.

In the experiment described below, we implement a grid search on kernels, the regularization term and gamma to find the optimal hyperparameters for the baseline SVM.

4.2 Experiments and results

The proposed CNN was tested against three datasets, namely, the Saarbrucken Voice Database (SVD) [67] the Spanish Parkinson's disease dataset (SPDD) [72] and the Arabic Voice Disorder Dataset (AVPD) [69,70]. We compare the classification accuracy of the proposed CNN with that of traditional machine learning algorithms including a Random Forest and a Support Vector Machine. Below we describe both the datasets and the results.

4.2.1 Datasets

The Saarbrucken Voice Database (SVD) [67] is a collection of speech samples from over 2000 people. There are three types of recordings in the dataset:

- sustained vowel sounds (/a/,/u/ and /i/) at normal, high and low pitch,
- sustained vowel sounds (/a/, /u/ and /i/) at rising-falling pitch,
- a conversational sentence in German.

In the experiment described below, we use a subset of SVD consisting of 50 pathological speech samples and 53 healthy speech samples of the sustained /a/ vowel. Multiple samples are extracted from each file.

The Spanish Parkinson's Disease Dataset (SPDD) [72] consists of speech samples from 50 Parkinson‘s disease patients and 50 healthy controls, 25 men and 25 women per group. All subjects are Colombian native Spanish speakers. Several types of speech recordings are included in the dataset:

- sustained vowels including /a/, /u/, /i/, /e/ and /o/,
- some specific words and phonemes,
- conversational speech.

As with SVD, we use the sustained vowel /a/ recordings to test our model and to compare its performance across two independent datasets.

The Arabic Voice Disorder Dataset (AVPD) [69, 70] is a collection of 350 normal and disordered subjects (175 healthy controls and 175 subjects with voice disorders). Three types of speech recordings are included:

- sustained vowel sounds (/a/, /u/ and /i/),
- isolated words including Arabic digits and common words,
- continuous speech.

As with SVD and SPDD, we also use sustained vowel /a/ sounds to test our CNN approach.

For all three datasets, we choose sustained vowel sounds instead of continuous speech to implement experiments with pathological speech classification since a) features extracted from sustained vowel sounds are computed in a more straightforward fashion compared with those computed from continuous speech [137] and b) sustained vowel sounds are independent of individual characteristics in human speech such as speaker dialects, speaking flow or linguistics which may affect the classification tasks [138]. Among several sustained vowels, we choose sustained vowel /a/ for all experiments since vowel sounds /a/ are available in almost all pathological speech datasets and the majority of research in pathological speech classification chooses the vowel /a/ for experiments [139]. Using the sustained vowel /a/ in our experiments enables us to obtain a directly comparable result with work in the field of pathological speech classification. The ratio of healthy to pathological speech data samples is 1 : 1 and thus, we choose accuracy as our evaluation metrics through all experiments described in Chapters 4, 5 and 6.

4.2.2 Experimental design

Spectrogram extraction

To extract spectrograms from raw speech in the dataset, we use the librosa [38] speech processing framework. Speech in the datasets is sampled at 44.1 kHz. The analysis windows length is 5.8 ms. The hop length is 1.4 ms. These speech processing parameters were chosen simply in order to obtain square spectrograms before feeding to classifiers. A Short-time Fourier Transform method is computed and 128 frequency components extracted from each window. 96 windows are analysed to give feature matrices of shape $(128, 96)$ which are then zero-padded to obtain $(128, 128)$ square matrices. Square spectrogram matrices are more convenient for use with a CNN classifier (avoiding issues with tensor sizes at intermediate values of the CNN).

Data splitting

We split the data into 80% for training and 20% for testing and apply such a split for all models evaluated in this thesis. The classification results are reported with accuracy evaluated on testing data.

The proposed CNN

The architecture of the CNN used in this experiment is summarised in Figure 4.2. The input layer has shape 128×128. The model contains three convolutional layers, one max-pooling layer, two fully-connected layers and one output layer organised as follows:

- The first convolutional layer has 16 filters of size 3 × 3, with a same padding and a stride of one followed by a batch normalization layer [140].
- The second convolutional layer has 32 filters of size 3 × 3, with a same padding and a stride of one followed by a batch normalization layer.
- The third convolutional layer has 64 filters of size 3 × 3, with a same padding and a stride of one followed by a batch normalization layer.
- A max-pooling layer with a size of two and a stride of two follows the convolutional layers and reduces the size of the data by a factor of two. This layer's output is flattened and fed into two fully-connected layers of 128 and 64 neurons respectively.
- The final output layer is a single neuron for binary classification with a sigmoid activation function. The total number of parameters is $1,638,113$. We use Keras on top of TensorFlow to build the model.

The CNN was trained using the Adam optimizer [96], the minibatch size was 32, the number of epochs was 30. The CNN was trained on 80% of each dataset and tested against the remaining 20% of that dataset.

Baseline algorithms

Our baseline RF model has a maximum depth decision tree of 20 and a number of trees of 10 and is constructed using the sklearn library [141]. We choose such maximum depth and number of trees by running a grid search to find the optimal hyperparameter setting. It serves as a baseline against which our CNN can be compared.

For the baseline SVM, in this experiment, we set the regularization parameter to be 10 and the gamma value to be 0.01. These parameters are also chosen based on a grid search to find the optimal hyperparameters. We also use the sklearn library [141] to build our SVM model.

To employ the RF and SVM as our baseline models, we first flatten the $(128, 128)$ spectrograms into $(16384, 1)$ vectors before feeding these vectors into the two models since they only accept one-dimensional data.

4.2.3 Results

Table 4.1: Results with sustained vowel /a/ sounds as input

Approach	SPDD	SVD	AVPD
RF	0.845	0.945	0.939
SVM	0.872	0.967	0.954
CNN	**0.967**	**0.980**	**0.987**

Table 4.2: Results with all vowel sounds as input

Approach	SPDD	SVD	AVPD
RF	0.735	0.796	0.845
SVM	0.725	0.849	0.876
CNN	**0.892**	**0.901**	**0.936**

The performance of three models (RF, SVM and CNN) across three datasets for the sustained vowel /a/ is summarised in Table 4.1. The results reported in this Table are based on an evaluation on the test set. Results show that the CNN model achieves 98% test accuracy on SVD, which is competitive with that reported by [81] (see Table 2.2). For all three datasets, our CNN outperforms traditional machine learning algorithms in terms of classification accuracy. The accuracy across all three models (RF, SVM and CNN) on the test set are slightly lower than that on the training set, indicating a slight overfitting with the models.

To further evaluate the robustness of the CNN approach, we test it against our baseline algorithms using a combination of other sustained vowel sounds (/a/, /u/, /i/, /e/ and /o/) as input data. The motivation behind this evaluation is that in a practical clinical setting, recorded speech samples may contain several sustained vowels from the same patients. We observe a drastic reduction (especially with SPDD) in classification accuracy with the baseline algorithms (see Table 4.2). With the CNN approach, we also observe a decrease in achieved accuracy; however, the CNN still achieves a reasonably high accuracy (around 90% with SPDD and SVD and 93.6% with AVPD). While acknowledging that the baseline algorithms (RF and SVM) require less time and energy to train, the accuracy gain due to the CNN is significant. The proposed CNN design achieves a high accuracy and is robust across several sustained vowel sounds while the classification accuracy achieved with baseline algorithms decreases significantly.

4.3 Summary

In this chapter we proposed a CNN for pathological speech classification. We tested our CNN-based approach across three commonly used pathological speech datasets: SPDD, SVD and AVPD. We achieved high accuracy classification with this approach and compared it with other traditional machine learning algorithms (Random Forest and Support Vector Machine). Our CNN was shown to outperform these discriminative algorithms in terms of classification accuracy. We subsequently tested the robustness of the CNN approach on a combined input data where we combined several sustained vowels to simulate a situation where multiple vowels are presented in patient data. We observed that while the accuracies achieved with baseline models (RF and SVM) reduced significantly, the accuracy achieved with the CNN approach also decreased but still remained acceptable high (around 90% with SVD and SPDD and 93.6% with AVPD). With our CNN now in place, in the next chapters we describe two generative-based semi-supervised learning (SSL) approaches where we aim to produce similar accuracy while relying on less labelled data.

You may say I'm a dreamer, but I'm not the only one. I hope someday you'll join us. And the world will live as one.

John Lennon

Chapter 5

Experiment #2: Semi-supervised Learning with Generative Adversarial Networks for Pathological Speech Classification

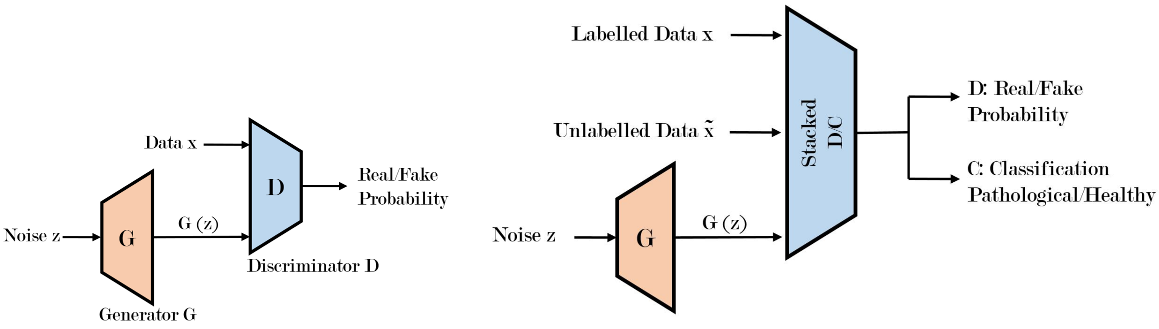
In this chapter, we seek to answer our second research question: *Can we employ generative models (GAN and VAE) to incorporate unlabelled data into the training process and thereby boost pathological speech classification accuracy?*. A challenge in applying machine learning algorithms to pathological speech classification (and to medical classification problems in general) is the labelled data shortage problem [27–30]. Labelled data acquisition often entails significant human effort and time-consuming experimental design. Further, for medical applications, privacy and ethical issues must be addressed where patient data is collected and stored. While labelled data are expensive and often scarce, unlabelled data are typically inexpensive and plentiful. In this chapter, we propose a semi-supervised learning approach that employs a generative adversarial network to incorporate both labelled and unlabelled data into training. We observe a promising accuracy gain with this approach compared to a baseline convolutional neural network trained only on labelled pathological speech data. The content of this chapter is published in Trinh N.H., O'Brien D. (2020) Generative Adversarial Network-Based Semi-supervised Learning for Pathological Speech Classification. In: Espinosa-Anke L., Martín-Vide C., Spasić I. (eds) Statistical Language and Speech Processing. SLSP 2020. Lecture Notes in Computer Science, vol 12379. Springer, Cham. The chapter is organised as follows: section 5.1 describes the methodology where we modify a GAN's discriminator to fit the task of semi-supervised learning; section 5.2 presents the experimental design and results; section 5.3 summarises our findings and concludes the chapter.

5.1 Methodology

In this section, we describe our modifications to the traditional GAN architecture to fit the task of semi-supervised pathological speech classification.

5.1.1 Architectural overview

The original GAN architecture [5] is illustrated in Figure 5.1a. A GAN is a generative model taking random noise as input and seeking to generate new samples from a real data distribution. A vanilla GAN consists of a discriminator and a generator. The generator takes random noise as input and generates new data samples. The discriminator's objective is to discriminate between real and generated samples (provided by the generator), classifying them as real or fake, respectively. The two networks compete until an equilibrium is reached where the discriminator cannot reliably discriminate between real and fake data. Let D be the discriminator and G be the generator. The minimax game between D and G is modelled mathematically as follows:



(a) The original GAN [5]

(b) The proposed Semi-Supervised GAN

Figure 5.1: Architecture overview

$$\min_{G}\max_{D} V(G,D) = E_{x \sim p_{data}(x)}[logD(x)] + E_{z \sim p_z(z)}[log(1 - D(G(z)))] \tag{5.1}$$

where $E_{x \sim p_{data}(x)}$ is the expected value over all real data samples with a data distribution $p_{data}(x)$, $D(x)$ is the probability that a real data sample is classified as real, $E_{z \sim p_z(z)}$ is the expected value over all noise samples with a prior noise distribution $p_z(z)$, $G(z)$ is the generated output from input noise z. The objective of the training process is to train D to maximize the probability of classifying generated samples $G(z)$ as fake and data samples x as real and to train G to convince D that generated samples, $G(z)$, are real. In other words, D is trained to maximize the loss function (5.1) while G is trained to minimize (5.1).

Semi-supervised GAN

To mitigate the problem of a shortage of training data, unlabelled and labelled data are incorporated into the training process in order to enhance the classification decision boundary (depicted in Figure 5.2). By incorporating unlabelled data, the semi-supervised model can shift the decision boundary to better cluster the data distribution [10]. This can be interpreted as the model attempting to first cluster the data and subsequently, identifying the decision boundary by assuming that unlabelled data points carry the same label as the labelled data region to which they most closely reside.

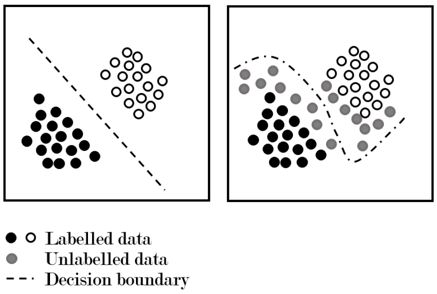


Figure 5.2: Data points in supervised learning with a limited amount of labelled data (left) and in semi-supervised learning with labelled data and unlabelled data (right) (from [10])

A GAN-based approach for semi-supervised learning (as illustrated in Figure 5.1b) incorporates data supplied from the GAN's generator and feeds the latter along with labelled and unlabelled data into the discriminator. In this work, we modify the discriminator to not only classify a data sample as real or fake (as in the original GAN formulation) but to also classify that sample as healthy or pathological. Following a similar method to that outlined in [37], we modify the discriminator's architecture by adding an additional output layer in parallel with the output layer handling real/fake classification to classify speech data as pathological or healthy. This can be considered as a stacking of a discriminator D (for real/fake discrimination) and a classifier C (for healthy/pathological classification).

As shown in Figure 5.4, the weights of the two networks (D and C) are shared across the input layer connected to the final hidden layer. Following the latter, the output layers of D and C are separated. A detailed description of this implementation is presented in section 5.2.2. The shared weight structure ensures that as D learns a feature representation from the unlabelled data, D shares that representation with C and helps C improve its feature learning compared with C being trained on only limited labelled data.

5.1.2 Loss functions

We train D to maximize the probability that D classifies both labelled data x and unlabelled data $\tilde{x}$ as real but generated data $G(z)$ as fake. We train C to classify the labelled data as healthy or pathological. We train G to maximize the probability that D will classify generated samples $G(z)$ as real. We derive the loss functions for D, C and G as follows:

$$Loss(D) = -(\mathbb{E}_{x \sim p_l(x)}[logD(x)] + \mathbb{E}_{\tilde{x} \sim p_u(\tilde{x})}[logD(\tilde{x})] + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))]) \quad (5.2)$$

$$Loss(C) = -\mathbb{E}_{(x,y) \sim p_l(x,y)}[ylogC(x)] \quad (5.3)$$

$$Loss(G) = \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))] \quad (5.4)$$

where $p_u(\tilde{x})$ and $p_l(x)$ are the unlabelled data and labelled distributions, $p_z(z)$ is the prior Gaussian noise distribution, $\mathbb{E}_{x \sim p_l(x)}$ is the expected value over all labelled data, $\mathbb{E}_{\tilde{x} \sim p_u(\tilde{x})}$ is the expected value over all unlabelled data, $\mathbb{E}_{z \sim p_z(z)}$ is the expected value over all noise samples, $\mathbb{E}_{(x,y) \sim p_l(x,y)}$ is the expected value over all labelled data points (x, y), $G(z)$ is the generated sample from the generator G, D is the probability that the discriminator classifies a data sample as real and $C(x)$ is the pathological/healthy classification result. The minimax game equation for the proposed semi-supervised GAN model is as follows:

$$\begin{aligned} \min_{G} \max_{D,C} J(G, D, C) = \mathbb{E}_{x \sim p_l(x)}[logD(x)] + \mathbb{E}_{\tilde{x} \sim p_u(\tilde{x})}[logD(\tilde{x})] \\ + \mathbb{E}_{(x,y) \sim p_l(x,y)}[ylogC(x)] \\ + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))] \end{aligned} \quad (5.5)$$

5.2 Experiments and results

In this section, we describe our experimental design and present our results.

5.2.1 Datasets

The contents of our three datasets, SPDD, SVD and AVPD have already been described. See section 4.2.1 for the details.

The Spanish Parkinson's Disease Dataset (SPDD)

We use speech data extracted from the sustained /a/ vowel recordings as labelled data and from other sustained vowel /e/, /i/, /o/ and /u/ as unlabelled data in the experiments described below.

The Saarbrucken Voice Database (SVD)

In our work, we make use of a subset of SVD data comprising of 50 pathological speech samples and 53 healthy speech samples from the sustained /a/ vowel as labelled data and sustained vowels /i/ and /u/ at different pitches (normal, high and low) as unlabelled data.

The Arabic Voice Disorder Dataset (AVPD)

Similar to SVD and SPDD, we also use sustained vowel /a/ samples as labelled data and sustained /i/ and /u/ vowels as unlabelled data.

For all three datasets, we include both healthy and pathological samples in the labelled and unlabelled sets. We choose to use sustained vowels /a/ as labelled data and other sustained vowels as unlabelled data to simulate a real-world situation in which the unlabelled speech data may not have the same vowel type as the labelled speech data, i.e. unlabelled and labelled data come from different distributions [142]. The purpose of the SSL approach is to extract useful features from alternative sustained vowels and to incorporate such features into learning a representation for sustained /a/ vowels. An additional reason for using the sustained /a/ vowel as labelled data is to obtain a directly comparable result with other work in the field where sustained

/a/ vowels were also extracted for pathological speech classification [81].

5.2.2 Experimental design

Speech Spectrogram Extraction

We follow the same approach as that in section 4.2.2.

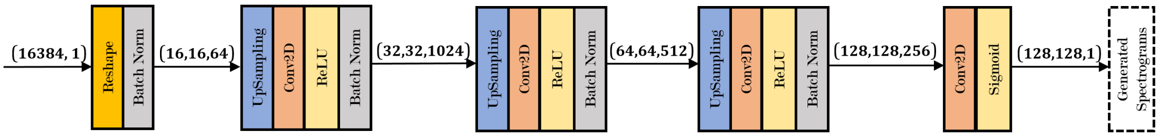


Figure 5.3: The generator

Semi-supervised GAN

The proposed semi-supervised GAN includes a stacked discriminator/classifier and a generator as shown in Figure 5.1b. Our GAN's architecture is inspired by that of the DCGAN [6]. The architectures of the generator and the discriminator are presented in Figures 5.3 and 5.4.

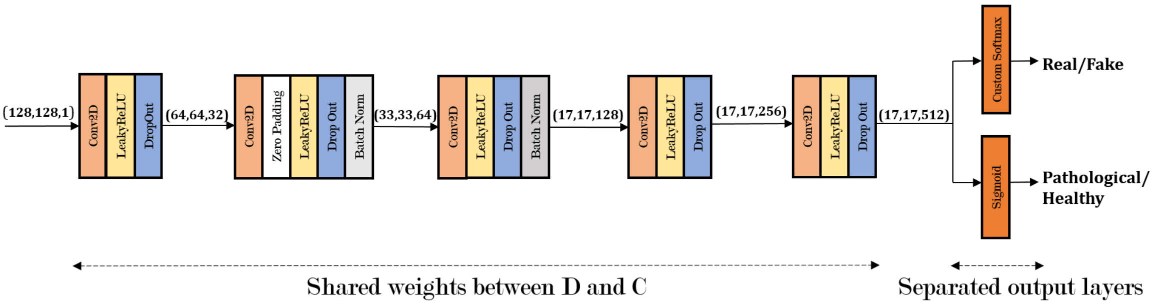


Figure 5.4: The stacked discriminator D and classifier C

The generator's architecture is depicted in Figure 5.3. The design of the generator is as follows:

- The generator's input is a Gaussian noise vector of shape $(16384, 1)$. The latter is reshaped to a square vector of shape $(16, 16, 64)$.

- Next, three stages of upsampling are applied to increase the data dimension from $(16, 16, 64)$ to $(128, 128, 256)$. Each stage includes an UpSampling layer followed by a convolutional layer with ReLU activation and a batch normalization layer [140].

- We finally apply a convolutional layer with a sigmoid activation function. The output of the generator is a tensor of shape $(128, 128, 1)$.

The discriminator's architecture is shown in Figure 5.4. The design of the discriminator is as follows:

- The input to the discriminator (the output of the generator) has shape $(128, 128, 1)$.

- We employ successive 2D convolutional layers with filter numbers of $32, 64, 128, 256$ and 512 respectively. To the output of each convolutional layer, we apply LeakyReLU with an alpha of 0.2, a drop-out layer with a rate of 0.25 and a batch normalization layer with a momentum of 0.8.

- The final output is flattened and a copy forwarded in two directions: to a discriminator for classification as fake or real and to a second classifier for pathological/healthy classification.

- For pathological speech classification, the final output layer is a single neuron with a sigmoid activation function for binary classification.

- For real/fake discrimination, we create a custom softmax layer to output the probability of the data being real.

Baseline CNN

To implement the baseline CNN, we reuse the GAN's discriminator architecture. This ensures results produced by the SSL and baseline approaches are comparable. The baseline is trained only on labelled data.

Training configuration

For each dataset, we train our models over 100 epochs, with a batch size of 32, with the Adam optimizer [96] and with a learning rate of 0.00002. Across models, we reduce the number of labelled spectrogram samples for training from $1,000$ to 800, 600, 400 and 200 and test on 800 spectrogram samples. We first randomly sample these spectrograms from the training set and test set and subsequently, train all models on the same set of spectrograms for each experiment to ensure that the classification results are directly comparable. We use $20,000$ unlabelled spectrograms (without healthy/pathological labels) as unlabelled data in the proposed SSL approach.

5.2.3 Results

In this section, we present, for visual inspection, several generated spectrograms and report the classification accuracy obtained with the GAN-based SSL approach.

Generative results

We present, for visual inspection, in Figure 5.5 sample spectrograms produced by the generator trained on the SPDD alongside some original spectrograms extracted from the same dataset. Similar frequency content is observed.

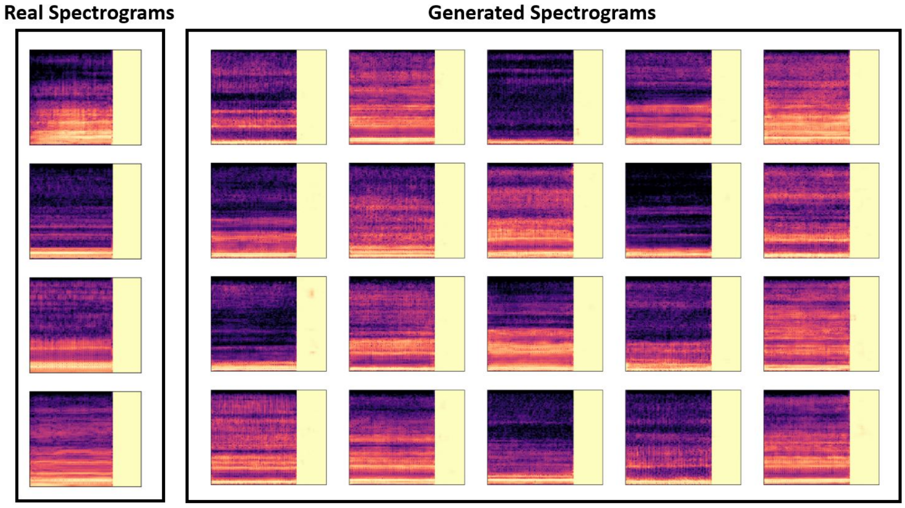


Figure 5.5: Original spectrograms (left) and generated spectrograms (right) using the proposed GAN

Classification accuracy

Accuracies obtained across the three datasets SPDD, SVD and AVPD are presented in Tables 5.1, 5.2 and 5.3 respectively. We compare the classification accuracy of the proposed semi-supervised GAN approach with that of the baseline CNN. We also compare the accuracy of the proposed approach against two additional classifiers previously proposed in the literature [81, 82]. We make several observations as follows:

- We observe an accuracy gain with the GAN-based SSL approach compared to baseline CNN models across all three datasets. The accuracy gains achieved with only 400 and 200 labelled data samples are promising across all three datasets.
- Among the three datasets, the GAN-based SSL approach shows the highest accuracy boost with SPDD. The specific reason requires further investigation.
- Among the baseline CNN models, the VGG16-based CNN yields the highest accuracy. This is because the VGG16-based CNN is a very large model with more than 138 million parameters [4].
- The GAN-based SSL trained on 600 labelled samples outperforms that trained on 1000 labelled samples. The specific reason for this behaviour requires further investigation.

Table 5.1: SPDD classification accuracy with the GAN-based SSL approach

Approach	Number of labelled data samples				
	1000	800	600	400	200
CNN [82]	0.896	0.835	0.851	0.798	0.705
VGG16-based CNN [81]	0.925	0.923	**0.929**	0.873	0.769
Baseline CNN	0.914	0.874	0.855	0.788	0.746
Proposed GAN-based SSL	**0.951**	**0.942**	0.919	**0.890**	**0.833**

Ablation study

To isolate the contribution of unlabelled data to the classification accuracy, we removed the latter data in an experiment using only the SPDD data to observe any drop in

Table 5.2: SVD classification accuracy with the GAN-based SSL approach

Approach	Number of labelled data samples				
	1000	800	600	400	200
CNN [82]	0.976	0.967	0.974	0.942	0.862
VGG16-based CNN [81]	**1.00**	**1.00**	0.993	0.984	0.946
Baseline CNN	1.00	0.998	0.985	0.973	0.939
Proposed GAN-based SSL	**1.00**	**1.00**	**0.999**	**0.998**	**0.960**

Table 5.3: AVPD classification accuracy with the GAN-based SSL approach

Approach	Number of labelled data samples				
	1000	800	600	400	200
CNN [82]	0.984	0.939	0.939	0.920	0.870
VGG16-based CNN [81]	**0.991**	0.991	0.978	0.963	0.860
Baseline CNN	0.990	0.986	0.966	0.944	0.818
Proposed GAN-based SSL	**0.991**	**0.998**	**0.993**	**0.971**	**0.889**

the classification performance. We select SPDD to implement the ablation study since experiments on SPDD show the largest classification boost, i.e. the most significant difference in accuracy between supervised CNN models and that of the semi-supervised GAN-based approach. The result of the ablation study is presented in Table 5.4. We observe a significant drop in the accuracy obtained, especially when training on only 400 and 200 labelled samples and without unlabelled data. This result further validates the positive effect on classification performance of applying semi-supervised learning to incorporate unlabelled data.

Table 5.4: SPDD ablation study with the GAN-based SSL approach

Proposed	Number of labelled data samples				
GAN-based SSL	1000	800	600	400	200
w/ unlabelled	**0.951**	**0.942**	**0.919**	**0.890**	**0.833**
w/o unlabelled	0.934	0.940	0.899	0.866	0.734

5.3 Summary

This chapter described a proposed GAN-based semi-supervised approach for pathological speech classification tasks. Results were presented that indicate the approach has the potential to mitigate the labelled data shortage problem faced by certain applications of deep learning. A GAN was incorporated into SSL by replacing the former's traditional binary discriminator with a multi-class discriminator that not only classified a sample as real or fake but also categorized that sample as healthy or pathological. We tested the approach against three commonly deployed pathological speech datasets: SPDD, SVD and AVPD. Comparing the performance of our GAN-based approach with a baseline CNN and two additional classifiers previously proposed in the literature [81,82], we observed a promising improvement in accuracy across models at each number of training samples when we decreased the amount of labelled training samples from 1000 through 800, 600, 400 and 200.

Future work will evaluate the performance of alternative GAN architectures (e.g. infoGAN [100] and marginGAN [129]) in semi-supervised pathological speech classification setting. Feature matching [37] could be explored as a means to improve discriminator performance. The proposed approach has potential applications not only in pathological speech classification but also across other audio classification tasks. In the following chapter we explore another application of generative modelling in the context of semi-supervised learning, namely representation learning, using a variational autoencoder and investigate whether the latter can be constructively employed in the pathological speech classification task.

Nobody ever figures out what life is all about, and it doesn't matter. Explore the world. Nearly everything is really interesting if you go into it deeply enough.

Professor Richard Feynman

Chapter 6

Experiment #3: Semi-supervised Learning with Variational Autoencoders for Pathological Speech Classification

In Chapter 5, we proposed a semi-supervised learning approach that employed a generative adversarial network to incorporate both unlabelled and labelled data into the training process. In this chapter, we continue to investigate our second research question (in this case we focus not on a GAN but on a VAE) and seek to also answer the third research question: *Which generative model (GAN or VAE) yields a better performance boost?* by testing whether another commonly used generative model – in this case a variational autoencoder (VAE)– can outperform a GAN in our semi-supervised setting. We first train a VAE on unlabelled data to extract useful features (representation learning) and subsequently, train a classifier employing the pre-trained encoder trained on labelled data with a fine-tune objective. We compare the classification accuracy of the VAE-based SSL approach with that of the GAN-based SSL approach. Results show that though our VAE-based approach outperforms the baseline CNN, the classification boost due to the VAE-based approach is less than that due to the GAN-based approach. A potential explanation for this trend is that while the GAN-based SSL approach concurrently employs two boosting strategies: representation learning and cluster-then-label, the VAE-based approach only incorporates unlabelled data for representation learning. The chapter is organised as follows: in section 6.1 we describe the VAE-based SSL approach; in section 6.2 we describe our experimental setting including datasets, neural network design and results; section 6.3 summarizes the chapter.

6.1 Methodology

In this section, we describe how we employ a VAE in an SSL approach, describing the architecture of the VAE, its loss function and the training process.

6.1.1 Architectural overview

Below we briefly describe the architecture of a vanilla VAE and how to employ a VAE in a two-step SSL training process.

Original VAE

The original VAE was discussed in detail in section 3.4.2 including an overview of its architecture and loss function. Here we briefly describe the architecture of the VAE depicted in Figure 3.12. A vanilla VAE includes an encoder and a decoder. Unlike a standard autoencoder, the VAE's encoder first encodes input data into two vectors: mean μ and standard deviation σ and subsequently, samples the latent space z by computing $z = \mu + \epsilon\sigma$ where ϵ is a normal distribution $\mathcal{N}(0, 1)$.

Semi-supervised learning with a VAE

To mitigate the labelled training data shortage, we employ a VAE for SSL as a representation learning model from unlabelled data by first training the VAE with unlabelled data and subsequently incorporating the VAE's pre-trained encoder component into a *fine-tuned CNN*. The fine-tuned CNN is trained on labelled data as described in Figure 6.1. Unlabelled data in the VAE approach is extracted using the same method as we employed in our previously described GAN setting: we extract spectrograms from sustained vowels /u/ and /i/ as unlabelled data to simulate a real-world situation arising in SSL tasks where unlabelled and labelled data come from different distributions. The goal of pre-training the VAE is to learn a feature representation from unlabelled data, in this context, to learn features from other sustained vowel sounds. This method of using a VAE for representation learning follows the M1 approach proposed in [36] where a VAE was employed to extract features from unlabelled data for SSL.

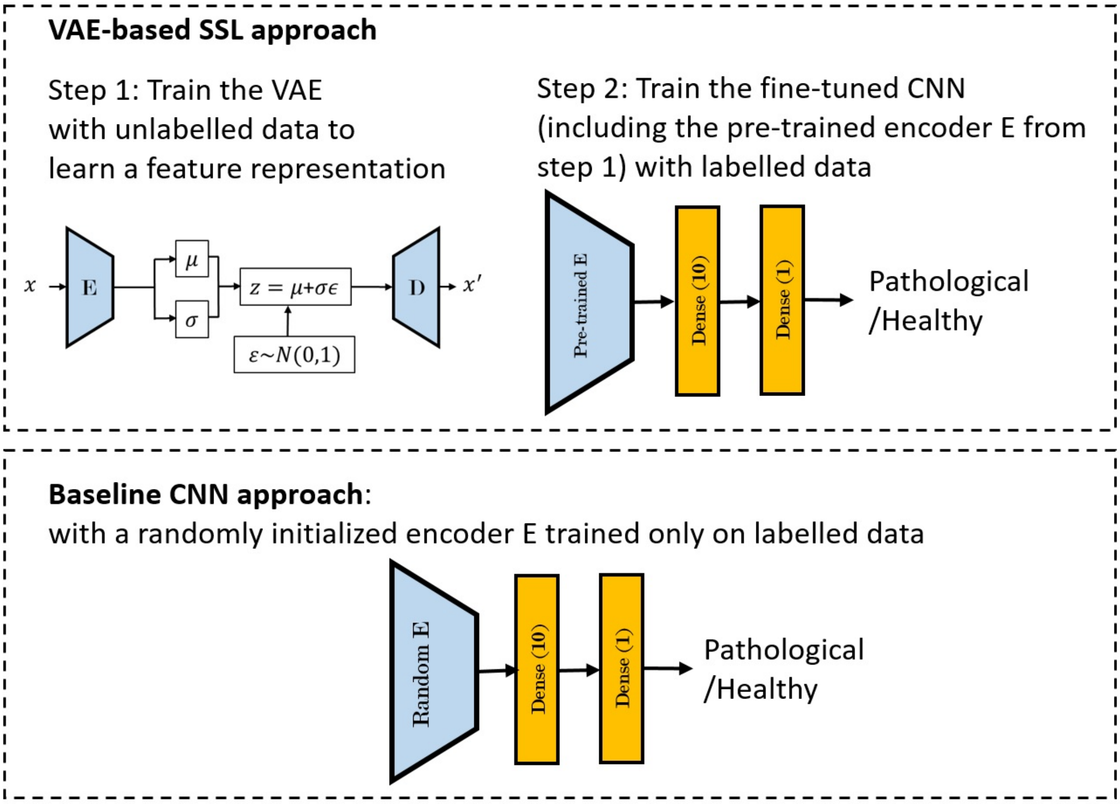

VAE-based SSL approach
Step 1: Train the VAE with unlabelled data to learn a feature representation
Step 2: Train the fine-tuned CNN (including the pre-trained encoder E from step 1) with labelled data
x
E
μ
σ
$z = \mu + \sigma\epsilon$
$\varepsilon \sim N(0,1)$
D
x'
Pre-trained E
Dense (10)
Dense (1)
Pathological /Healthy
Baseline CNN approach:
with a randomly initialized encoder E trained only on labelled data
Random E
Dense (10)
Dense (1)
Pathological /Healthy

Figure 6.1: VAE-based approach for semi-supervised learning

6.1.2 Loss function

We set the loss function of the VAE as in a typical VAE:

$$Loss(\phi, \theta) = \mathbb{E}_{q_\theta(z|x)}[logp_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p_\theta(z)) \tag{6.1}$$

where θ and ϕ are the parameters of the encoder and decoder networks. The first term in equation (6.1) is the reconstruction loss representing the difference between input and output whereas the second term is the regularization term representing the KL divergence between the latent space at the output of the encoder and a univariate Gaussian distribution (see section 3.4.2 in Chapter 3 for more details).

6.2 Experiments and results

In this section, we describe the datasets used and present the design architecture of our variational autoencoder. We discuss the experimental results and compare the classification accuracy of the proposed VAE approach with that obtained by the baseline CNN and the GAN-based SSL approach.

6.2.1 Datasets

For our VAE-based SSL experiments, we use the same data as for our GAN-based SSL experiments (see section 5.2.1). For the details on the dataset contents, see section 4.2.1.

Similar to the GAN-based experiment described in Chapter 5, we also extract sustained /a/ vowels as labelled data and other sustained vowels /u/ and /i/ as unlabelled data to simulate a real-world situation arising frequently with SSL tasks where the unlabelled data carry some useful information about the labelled data but may not be of the same type as the labelled data, i.e unlabelled and labelled data may come from different distributions [142]. In a pathological speech SSL task, labelled data can be one type of sustained vowel but unlabelled data may be in different types of sustained vowel sounds.

6.2.2 Experimental design

Spectrogram extraction

We follow the same approach as that in section 4.2.2.

Variational autoencoder

For this experiment, we designed and implemented a VAE with the architecture shown in Figures 6.2, 6.3 and 6.4. Our proposed VAE includes three main components: an encoder (see Figure 6.2), a sampling layer and a decoder (see Figure 6.3).

The encoder design is depicted in Figure 6.2 and is organised as follows:

- The input to the encoder is a spectrogram of shape $(128, 128, 1)$. We employ four successive 2D convolutional layers with filter numbers of $32, 64, 64$ and 64 respectively.

- To the output of each convolutional layer, we apply ReLU activation, a batch normalization layer and a two-by-two max-pooling layer. The output after four convolution stages has shape $(8, 8, 64)$.

- We flatten this output to shape $(4092, 1)$ and apply a fully-connected network of 512 neurons followed by a drop-out layer with a rate of 0.3.

- The output of the encoder (the latent lower dimensional feature representation) is a vector of shape $(512, 1)$.

The decoder design is depicted in Figure 6.3 and is as follows:

- The input to the decoder has shape $(512, 1)$. We employ a fully-connected layer with 256 neurons and reshape the output of this layer to obtain a tensor with shape $(16, 16, 1)$.

- We employ three up-sampling layers to successively double the size of the tensor from 16 × 16 to 32 × 32, to 64 × 64 to 128 × 128. To the output of each up-sampling layer, we apply a convolutional layer, a ReLU activation and batch normalization layer.

- We finally employ a convolutional layer followed by a sigmoid activation to reduce the channel size from 256 to 1. The output of the decoder is of shape $(128, 128, 1)$.

The high-level architecture of the overall VAE is depicted in Figure 6.4. After the encoder compresses input data from shape $(128, 128, 1)$ to a vector of shape $(512, 1)$, the output of the encoder is fed into the sampling layer by separating into two fully-connected layers to produce two vectors μ and σ. An ϵ vector is then sampled from a univariate Gaussian distribution. The output of the sampling layer is calculated as $z = \mu + \sigma\epsilon$. In the literature, this is referred to as ”the reparameterization trick” where the goal is to make the VAE's sampling layer differentiable for gradient-based optimization [9]. The output z also has shape $(512, 1)$ and is fed into the decoder to generate new spectrograms.

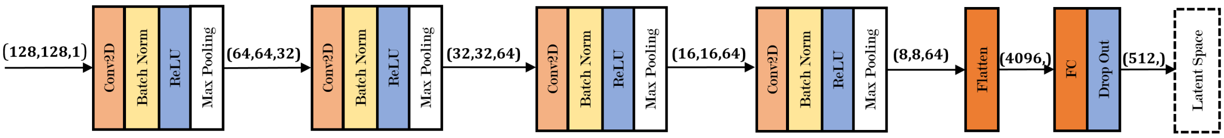


Figure 6.2: Architecture of the encoder

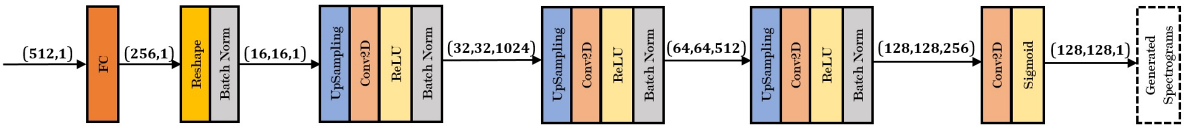


Figure 6.3: Architecture of the decoder

Fine-tuned CNN with the pre-trained encoder

After pre-training the VAE, we incorporate the pre-trained encoder into a fine-tuned CNN. The architecture of the fine-tuned CNN is depicted in Figure 6.5. The output of

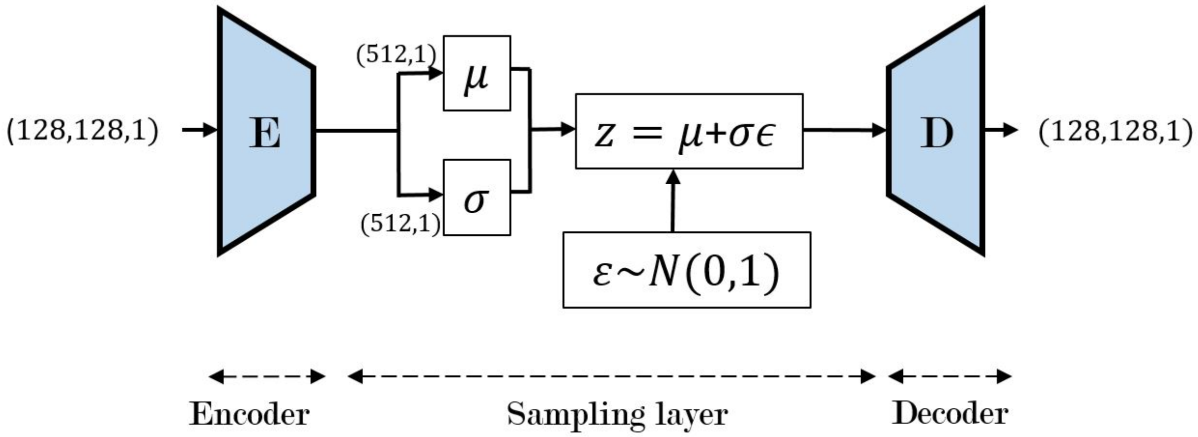


Figure 6.4: Variational autoencoder for spectrogram generation

the pre-trained encoder is fed into two fully-connected layers of sizes ten and one. The ten-neuron layer is added before the final output layer to reduce overfitting instead of connecting all 512 neurons to just one neuron at the output layer. The output layer is a sigmoid activation function with an output probability that a speech sample is pathological.

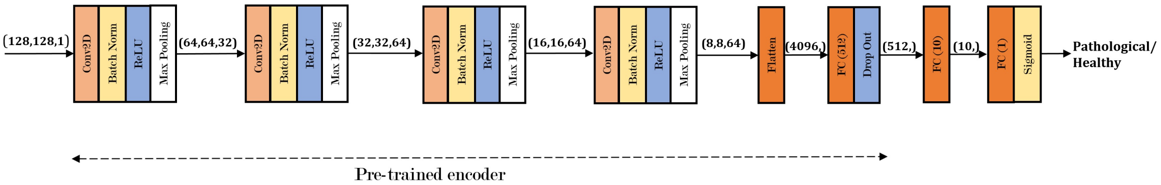


Figure 6.5: Architecture of the fine-tuned CNN with the pre-trained encoder

Baseline CNN

For the baseline, we employ a CNN with the same architecture (as depicted in Figure 6.6) as that of the fine-tuned CNN; however, we do not incorporate the pre-trained encoder into the baseline CNN. Instead, we employ random initialization for the baseline CNN and subsequently train this CNN only on labelled data. Notice that we allow the training to update the randomly initialized encoder.

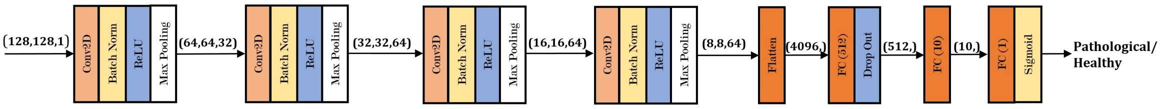


Figure 6.6: Architecture of the baseline CNN without the pre-trained encoder

Training configuration

We first train the VAE with unlabelled spectrogram data extracted from sustained vowel speech samples /u/ and /i/ at different pitches in 20 epochs with $20,000$ unlabelled data points. We then incorporate the pre-trained encoder into the supervised CNN and train this CNN on labelled data extracted from sustained /a/ speech samples in 100 epochs. For the baseline CNN, we train only on labelled data in 100 epochs. In all training processes, we employ the Adam optimizer [96] with a learning rate of 0.00002. Across experiments, we reduce the number of labelled spectrogram samples for training from $1,000$ to 800, 600, 400 and 200 and test on 800 spectrogram samples. The ratio of healthy to pathological speech data points is $1:1$.

6.2.3 Results

Below we present several generated spectrograms for visual inspection and the classification accuracy obtained with the VAE-based SSL approach.

Generative results

We present in Figure 6.7, for visual inspection, several sample spectrograms generated using the VAE-based approach. We observe similar frequency content across the randomly sampled spectrograms; however, blurriness can be observed in these spectrograms compared to those produced by the generator in the GAN-based SSL experiment described in Chapter 5 (see Figure 5.5). This blurriness is expected as the lack of definition stems from their attempt to regularize a Gaussian distribution encoding the latent space. As a result, many data points in the training data can have the same encoding z in the latent space [143,144], leading to blurriness in images generated by VAEs [5].

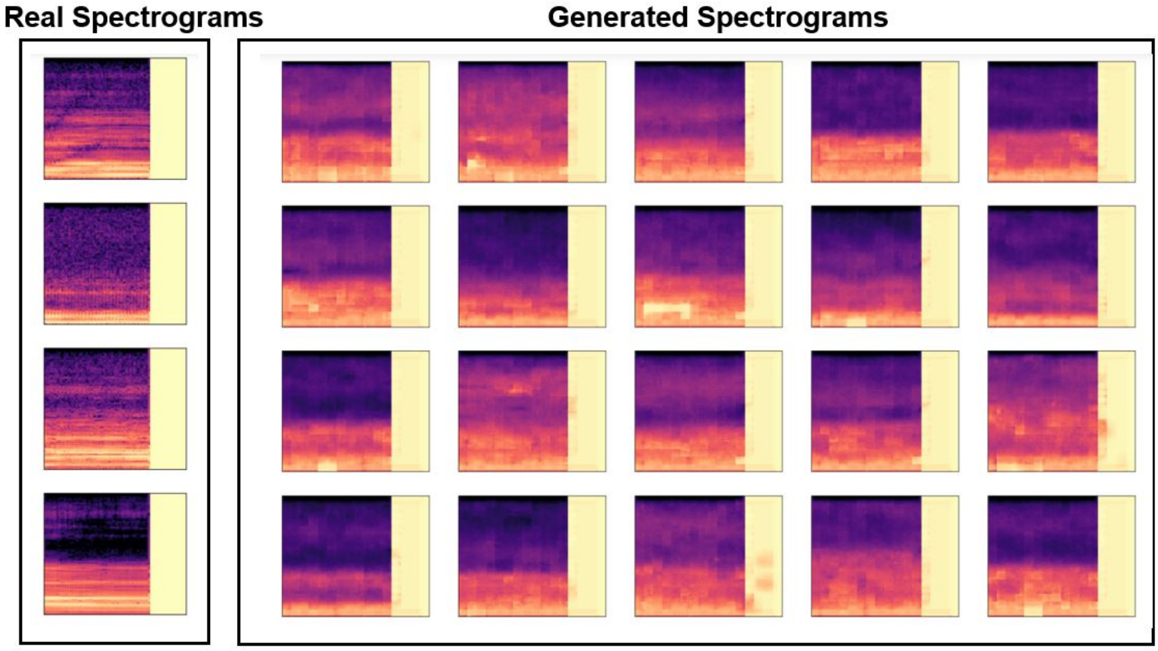


Figure 6.7: Original spectrograms (left) and generated spectrograms (right) using the proposed VAE

Classification results

We present in Tables 6.1, 6.2 and 6.3 the classification accuracy results with SPDD, SVD and AVPD using the VAE-based SSL approach. We make several observations as follows:

- Classification results across three pathological speech datasets show that the VAE-based SSL outperforms the baseline CNN, especially when the number of labelled data samples reduces to 400 and 200. With 200 labelled data samples of SPDD, while the baseline CNN only achieves 70.7% accuracy, the VAE-based approach still achieves 78.2% accuracy. This implies that the VAE-based approach can boost the classification accuracy by learning a feature representation from unlabelled data.

- With SPDD, the performance of the VAE-based approach is comparable with that of the VGG16-based CNN. With SVD and AVPD, the VGG16-based CNN outperforms the VAE-based approach. This may be due to the fact that the VGG16-based CNN is a very large model with more than 138 million parameters [4] while the number of parameters in the proposed VAE is 8.24 million parameters.

- The GAN-based SSL approach outperforms the VAE-based SSL approach. A potential explanation for this trend is that the GAN-based SSL approach combines two SSL boosting strategies: representation learning and cluster-then-label strategies when feeding the unlabelled data points to the stacking discriminator (see Section 5.1 for more details) whereas the VAE-based SSL approach only boosts the classification performance by learning a feature representation from the unlabelled data.
- Among the three datasets, the VAE-based and GAN-based SSL approaches show the highest classification boost on SPDD. The specific reason requires further investigation.

Table 6.1: SPDD classification accuracy with the VAE-based SSL approach

Approach	Number of labelled data samples				
	1000	800	600	400	200
CNN [82]	0.896	0.835	0.851	0.798	0.705
VGG16-based CNN [81]	0.925	0.923	**0.929**	0.873	0.769
Baseline CNN	0.890	0.854	0.851	0.792	0.707
Proposed GAN-based SSL	**0.951**	**0.942**	0.919	**0.890**	**0.833**
Proposed VAE-based SSL	0.931	0.938	0.894	0.814	0.782

Ablation study

We study the improvements attributable to the use of unlabelled data by removing the latter when training the VAE. Here we pre-train the VAE only with labelled data for initializing a set of parameters for the VAE. Thus, the VAE learns features from a limited number of /a/ samples rather than learning from a mix of sustained

Table 6.2: SVD classification accuracy with the VAE-based SSL approach

Approach	Number of labelled data samples				
	1000	800	600	400	200
CNN [82]	0.976	0.967	0.974	0.942	0.862
VGG16-based CNN [81]	**1.00**	**1.00**	0.993	0.984	0.946
Baseline CNN	0.995	0.996	0.965	0.956	0.910
Proposed GAN-based SSL	**1.00**	**1.00**	**0.999**	**0.998**	**0.960**
Proposed VAE-based SSL	0.998	0.995	0.989	0.985	0.910

Table 6.3: AVPD classification accuracy with the VAE-based SSL approach

Approach	Number of labelled data samples				
	1000	800	600	400	200
CNN [82]	0.984	0.939	0.939	0.920	0.870
VGG16-based CNN [81]	0.991	0.991	0.978	0.963	0.860
Baseline CNN	0.979	0.954	0.944	0.924	0.778
Proposed GAN-based SSL	0.991	**0.998**	**0.993**	**0.971**	**0.889**
Proposed VAE-based SSL	**0.995**	0.990	0.948	0.943	0.842

vowels. The ablation study results are presented in Table 6.4. We observe a decrease in performance when removing unlabelled data from the VAE training. This result further validates the positive effect of unlabelled data on improving the classification performance of the fine-tuned CNN.

Table 6.4: SPDD ablation study with the VAE-based SSL approach

Proposed VAE-based SSL	Number of labelled data samples				
	1000	800	600	400	200
w/ unlabelled	**0.931**	**0.938**	**0.894**	**0.846**	**0.782**
w/o unlabelled	0.911	0.894	0.875	0.824	0.762

6.3 Summary

In this chapter, we proposed a VAE-based SSL approach in which we employed unlabelled data in VAE training. The latter's pre-trained encoder was incorporated into the training of a fine-tuned CNN on labelled data. We compared the performance of the VAE-based SSL approach with that of a baseline CNN. We observed an accuracy gain with the VAE-based SSL approach across all three pathological speech datasets.

However, the classification accuracy of the VAE-based SSL approach was lower than that of the GAN-based SSL approach. This can be explained as the GAN-based SSL approach combines and leverages two boosting strategies: representation learning and cluster-then-label when feeding the unlabelled data points to the stacking discriminator. By contrast, the VAE-based SSL approach boosts the classification performance by learning a feature representation only from the unlabelled data. To isolate the classification gain due to unlabelled data, in an additional experiment, we used only limited labelled data for the VAE training. We observed a corresponding decrease in performance.

Future work will explore alternative recently proposed VAE architectures (e.g. β-VAE [145], infinite VAE [146] or Coupled VAE [147]). Other improvements could be achieved by altering the VAE architecture, e.g. the size of latent space z and the design of the encoder and decoder.

As a technologist, I see how AI and the fourth industrial revolution will impact every aspect of people's lives.

Professor Fei-Fei Li

Chapter 7

Conclusion

This chapter concludes the thesis by highlighting our contributions, noting any limitations of the work and suggesting areas for future investigation. The chapter is organised as follows: section 7.1 reviews the thesis; section 7.2 notes several limitations and corresponding future work; section 7.3 provides a final remark.

7.1 Review

We first review, in Chapter 2, related work in pathological speech classification. A challenge in pathological speech classification (and a general problem for clinical applications of deep learning) is the data shortage problem due to the significant human effort required to acquire and label patient speech data. Speech data collection also raises privacy and ethical issues. In Chapter 3 we provided the background material in deep learning that underpins the experiments presented in subsequent chapters. The background material included an overview of deep neural networks, convolutional neural networks, generative adversarial networks and variational autoencoders.

Chapters 4-6 dealt with our research questions. In Chapter 4 we described our implementation of a convolutional neural network for pathological speech classification trained only on labelled data. To mitigate the data shortage problem, we presented in Chapter 5 our design and implementation of a semi-supervised learning approach based on a generative adversarial network. In Chapter 6, we describe an alternative SSL approach based on a variational autoencoder.

7.1.1 Research questions

Research question 1: *Can deep neural networks outperform traditional machine learning algorithms in pathological speech classification?*

In Chapter 4, we described an experiment using a CNN for pathological speech classification across three commonly used datasets: the Spanish Parkinson Disease Datasets (SPDD), the Saarbrucken Voice Database (SVD) and the Arabic Voice Pathological Database (AVPD). We compared the performance of the CNN with that of an RF and SVM. We observed high accuracy across all three datasets with the proposed CNN (see Table 4.1). We conclude that the CNN outperforms traditional machine learning algorithms in terms of classification accuracy across single and mixed vowels. Using the proposed CNN approach, we achieved an SVD accuracy comparable the state-of-the-art work reported in [81].

Research question 2: *Can we employ generative models (GAN and VAE) to incorporate unlabelled data into the training process and thereby boost pathological speech classification accuracy?*

To answer research question 2, we implemented two SSL approaches using two commonly used generative models, the GAN and VAE, to incorporate unlabelled data into the training process with the goal of boosting classification accuracy.

We described our GAN-based SSL approach in Chapter 5. To incorporate unlabelled data using a GAN, we modified the discriminator by stacking the classifier for healthy/pathological classification and the discriminator for real/fake classification. The purpose of this modification was to share weights between the discriminator and the classifier such that the feature representation learned by the discriminator from unlabelled data is shared with the classifier trained only on limited labelled data. With this approach, we achieved an accuracy gain compared to the supervised CNN approach trained only on labelled data (see Tables 5.1, 5.2 and 5.3). We also implemented an ablation study to quantify the contribution of unlabelled data by removing it when training the proposed GAN. We observed a drop in performance when training without unlabelled data (see Table 5.4). This highlights the positive effect on classification accuracy of incorporating unlabelled data into the training process.

We described our VAE-based SSL approach in Chapter 6. We incorporated unlabelled data by first training a traditional VAE with unlabelled data with a loss function including a reconstruction loss and a regularization loss. We subsequently incorporated the pre-trained encoder into a supervised CNN trained on labelled data. We also observed an accuracy gain with the proposed VAE-based approach compared to the baseline CNN trained only on labelled data (see Tables 6.1, 6.2 and 6.3). Again, we implemented an ablation study in which we removed unlabelled data from the VAE pre-training process. We observed a drop in performance in this study (see Table 6.4) demonstrating that unlabelled data contributed to the improvement in classification accuracy.

Research question 3: *Which generative model (GAN or VAE) yields a better performance boost?*

We compared the performance of the GAN-based SSL and the VAE-based SSL approaches (see Tables 6.1, 6.2 and 6.3) in terms of classification accuracy. The GAN-based SSL approach outperformed the VAE-based approach. A potential explanation for this better performance is that the GAN in an SSL setting combines and leverages two boosting strategies: representation learning and cluster-then-label while the VAE-based approach boosts the classification accuracy by learning only a feature representation from unlabelled data.

7.1.2 Contributions

Our contributions are as follows:

- In Chapter 2 we presented a literature review of previous work in pathological speech classification that identified key papers and databases useful for those working in the field.

- In Chapter 4 we proposed a CNN architecture and trained it on labelled data for pathological speech classification. This proposed CNN outperformed the traditional machine learning algorithms (Random Forest and Support Vector Machine) in terms of classification accuracy across all three datasets.
- In Chapter 5 we proposed, designed and implemented a GAN-based SSL approach for pathological speech classification. To incorporate unlabelled data into the training, we stacked the discriminator and the classifier with the goal of parameter sharing, i.e. when the discriminator learns a feature representation from unlabelled data, it also shares these learned features with the classifier. Our experimental results demonstrated the proposed approach reduces dependency on labelled data while maintaining high accuracy.
- In Chapter 6 we proposed, designed and implemented a VAE-based SSL approach for pathological speech classification by first training a VAE with unlabelled data and then incorporating the pre-trained encoder into a fine-tuned CNN. Our experimental results demonstrated a corresponding performance boost. We compared the performance of the GAN-based approach with that of the VAE-based approach

7.2 Limitations and future work

We present below four limitations in this work and describe how they might be addressed in future work.

- **Limitation 1:** We only employ vanilla GAN and VAE architectures. Future work will experiment with alternative GAN (e.g. info GAN [100] and marginGAN [129]) and VAE (e.g. β-VAE [145], infinite VAE [146] or Coupled

VAE [147]) designs. Other neural network architectures for the encoder and the decoder in the VAE and for the discriminator and the generator in the GAN could be examined to find the most appropriate architectures for the proposed approaches. Hyperparameter tuning using grid search could also be applied to find the best hyperparameters for each model.

- **Limitation 2:** Another limitation is a lack of data visualisation to provide an intuitive explanation of how the proposed SSL approaches improve the overall classification performance. Data visualisation will better explain how the proposed approaches improved the classification accuracy. Visualisation methods can be implemented using dimensionality reduction techniques, e.g t-Stochastic Neighboring Embedding (t-SNE) [148] and Principal Component Analysis (PCA) [87]. Visualising the activations of neurons in hidden layers [149–152] will also provide a better understanding of how the neural networks learn a feature representation from both unlabelled data and labelled data.

- **Limitation 3:** Due to limited time and computing resources, we used the random train/test split approach, i.e. we trained on 80% of the dataset and tested against the remaining 20%. Future work will use k-fold cross-validation method by dividing the data into k folds, training on the first $k-1$ folds and validating on the k^{th} fold.

- **Limitation 4:** Alternative speech processing parameters for spectrogram calculation are worth investigation (e.g. window length, window type, step size, sampling rate (22kHz/16kHz/8kHz)) in order to measure their effect on classification performance. It would be interesting to explore feature representations apart from the spectrogram. A host of alternative features have been proposed in the literature incorporating MFCCs, glottal features, voicing, harmonic-to-noise ratios, etc.

7.3 Final remarks

In medical applications the lack of labelled data imposes a major challenge on early disease diagnosis and classification. In this work we demonstrated that semi-supervised learning approaches using generative models have the potential to alleviate this problem at least for pathological speech classification. We believe however that the approach has potential application beyond pathological speech classification in the audio classification and medical fields.

Bibliography

[1] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[2] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[6] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[7] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[8] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[9] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[10] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.

[11] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[12] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248, 2017.

[13] Daniele Ravì, Charence Wong, Fani Deligianni, Melissa Berthelot, Javier Andreu-Perez, Benny Lo, and Guang-Zhong Yang. Deep learning for health informatics. *IEEE journal of biomedical and health informatics*, 21(1):4–21, 2016.

[14] Linda Wang and Alexander Wong. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *arXiv preprint arXiv:2003.09871*, 2020.

[15] Ali Narin, Ceren Kaya, and Ziynet Pamuk. Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. *arXiv preprint arXiv:2003.10849*, 2020.

[16] Hojjat Salehinejad, Shahrokh Valaee, Tim Dowdell, Errol Colak, and Joseph Barfett. Generalization of deep neural networks for chest pathology classification in x-rays using generative adversarial networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 990–994. IEEE, 2018.

[17] Yanbo Zhang and Hengyong Yu. Convolutional neural network based metal artifact reduction in x-ray computed tomography. *IEEE transactions on medical imaging*, 37(6):1370–1381, 2018.

[18] Cosmin Cernazanu-Glavan and Stefan Holban. Segmentation of bone structure in x-ray images using convolutional neural network. *Adv. Electr. Comput. Eng*, 13(1):87–94, 2013.

[19] Xiaojun Zhang, Zhi Tao, Heming Zhao, and Tianqi Xu. Pathological voice recognition by deep neural network. In *2017 4th International Conference on Systems and Informatics (ICSAI)*, pages 464–468. IEEE, 2017.

[20] Z. Chuang, X. Yu, J. Chen, Y. Hsu, Z. Xu, C. Wang, F. Lin, and S. Fang. Dnn-based approach to detect and classify pathological voice. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 5238–5241, Dec 2018.

[21] P. Harar, J. B. Alonso-Hernandezy, J. Mekyska, Z. Galaz, R. Burget, and Z. Smekal. Voice pathology detection using deep learning: a preliminary study. In *2017 International Conference and Workshop on Bioinspired Intelligence (IWOBI)*, pages 1–4, July 2017.

[22] D. Huang, M. Dong, and H. Li. Combining multiple kernel models for automatic intelligibility detection of pathological speech. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6485–6489, March 2016.

[23] Meysam Asgari and Izhak Shafran. Extracting cues from speech for predicting severity of parkinson's disease. In *2010 IEEE International Workshop on Machine Learning for Signal Processing*, pages 462–467. IEEE, 2010.

[24] Athanasios Tsanas, Max A Little, Patrick E McSharry, Jennifer Spielman, and Lorraine O Ramig. Novel speech signal processing algorithms for high-accuracy classification of parkinson's disease. *IEEE transactions on biomedical engineering*, 59(5):1264–1271, 2012.

[25] Betul Erdogdu Sakar, M Erdem Isenkul, C Okan Sakar, Ahmet Sertbas, Fikret Gurgen, Sakir Delil, Hulya Apaydin, and Olcay Kursun. Collection and analysis of a parkinson speech dataset with multiple types of sound recordings. *IEEE Journal of Biomedical and Health Informatics*, 17(4):828–834, 2013.

[26] Amir Hossein Poorjam, Max A Little, Jesper Rindom Jensen, and Mads Græsbøll Christensen. A parametric approach for classification of distortions in pathological voices. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 286–290. IEEE, 2018.

[27] Rahul C Deo. Machine learning in medicine. *Circulation*, 132(20):1920–1930, 2015.

[28] Christoph Baur, Shadi Albarqouni, and Nassir Navab. Semi-supervised deep learning for fully convolutional networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 311–319. Springer, 2017.

[29] Wenqing Sun, Tzu-Liang Bill Tseng, Jianying Zhang, and Wei Qian. Enhancing deep convolutional neural network scheme for breast cancer diagnosis with unlabeled data. *Computerized Medical Imaging and Graphics*, 57:4–9, 2017.

[30] Dong Nie, Yaozong Gao, Li Wang, and Dinggang Shen. Asdnet: Attention based semi-supervised deep networks for medical image segmentation. In *International*

Conference on Medical Image Computing and Computer-Assisted Intervention, pages 370–378. Springer, 2018.

[31] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.

[32] P Kingma Diederik, Max Welling, et al. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, volume 1, 2014.

[33] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[34] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[35] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[36] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.

[37] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[38] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, 2015.

[39] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.

[40] Gregory Hickok. Computational neuroanatomy of speech production. *Nature reviews neuroscience*, 13(2):135–145, 2012.

[41] Nina Dronkers and Jennifer Ogar. Brain areas involved in speech production, 2004.

[42] Nina F Dronkers, Odile Plaisant, Marie Therese Iba-Zizen, and Emmanuel A Cabanis. Paul broca's historic cases: high resolution mr imaging of the brains of leborgne and lelong. *Brain*, 130(5):1432–1441, 2007.

[43] Jacob Benesty, M Mohan Sondhi, and Yiteng Huang. *Springer handbook of speech processing*. Springer, 2007.

[44] Tom Bäckström. *Speech coding: With code-excited linear prediction*. Springer, 2017.

[45] Bryan Gick, Ian Wilson, and Donald Derrick. *Articulatory phonetics*. John Wiley & Sons, 2012.

[46] Dario Marcotullio, Giuseppe Magliulo, and Tiziana Pezone. Reinke's edema and risk factors: clinical and histopathologic aspects. *American journal of otolaryngology*, 23(2):81–84, 2002.

[47] Steven M Zeitels, Glenn W Bunting, Robert E Hillman, and Traci Vaughn. Reinke's edema: phonatory mechanisms and management strategies. *Annals of Otology, Rhinology & Laryngology*, 106(7):533–543, 1997.

[48] Steven M Zeitels, Roy R Casiano, Glendon M Gardner, Norman D Hogikyan, James A Koufman, and Clark A Rosen. Management of common voice problems: Committee report. *Otolaryngology—Head and Neck Surgery*, 126(4):333–348, 2002.

[49] C Pototschnig and WF Thumfart. Electromyographic evaluation of vocal cord disorders. *Acta oto-rhino-laryngologica belgica*, 51(2):99–104, 1997.

[50] Lucian Sulica and Andrew Blitzer. Vocal fold paresis: evidence and controversies. *Current Opinion in Otolaryngology & Head and Neck Surgery*, 15(3):159–162, 2007.

[51] Mausumi N Syamal and Michael S Benninger. Vocal fold paresis: a review of clinical presentation, differential diagnosis, and prognostic indicators. *Current Opinion in Otolaryngology & Head and Neck Surgery*, 24(3):197–202, 2016.

[52] Lucian Sulica. Vocal fold paresis: an evolving clinical concept. *Current Otorhinolaryngology Reports*, 1(3):158–162, 2013.

[53] Katherine Verdolini, Clark A Rosen, and Ryan C Branski. Vocal fold nodules (nodes, singer's nodes, screamer's nodes. *Classification Manual for Voice Disorders-I*, pages 37–40, 2014.

[54] M Civera, CM Filosi, NM Pugno, M Silvestrini, C Surace, K Worden, et al. Assessment of vocal cord nodules: a case study in speech processing by using hilbert-huang transform. In *J. Phys. Conf. Ser*, volume 842, page 012025, 2017.

[55] Beata Miaśkiewicz, Agata Szkiełkowska, Elżbieta Gos, Aleksandra Panasiewicz, Elżbieta Włodarczyk, and Piotr H Skarżyński. Pathological sulcus vocalis: treatment approaches and voice outcomes in 36 patients. *European Archives of Oto-Rhino-Laryngology*, 275(11):2763–2771, 2018.

[56] Charles N Ford, Katsuhide Inagi, Aliaa Khidr, Diane M Bless, and Kennedy W Gilchrist. Sulcus vocalis: a rational analytical approach to diagnosis and management. *Annals of otology, rhinology & laryngology*, 105(3):189–200, 1996.

[57] Pamela Enderby. Frenchay dysarthria assessment. *British Journal of Disorders of Communication*, 15(3):165–173, 1980.

[58] Susan B O'Sullivan, Thomas J Schmitz, and George Fulk. *Physical rehabilitation*. FA Davis, 2019.

[59] Joseph R Duffy. *Motor Speech Disorders E-Book: Substrates, Differential Diagnosis, and Management*. Elsevier Health Sciences, 2019.

[60] Werner Poewe, Klaus Seppi, Caroline M Tanner, Glenda M Halliday, Patrik Brundin, Jens Volkmann, Anette-Eleonore Schrag, and Anthony E Lang. Parkinson disease. *Nature reviews Disease primers*, 3(1):1–21, 2017.

[61] Alastair J Noyce, Jonathan P Bestwick, Laura Silveira-Moriyama, Christopher H Hawkes, Gavin Giovannoni, Andrew J Lees, and Anette Schrag. Meta-analysis of early nonmotor features and risk factors for parkinson disease. *Annals of neurology*, 72(6):893–901, 2012.

[62] Suzanne Lesage and Alexis Brice. Parkinson's disease: from monogenic forms to genetic susceptibility factors. *Human molecular genetics*, 18(R1):R48–R59, 2009.

[63] Douglas J Gelb, Eugene Oliver, and Sid Gilman. Diagnostic criteria for parkinson disease. *Archives of neurology*, 56(1):33–39, 1999.

[64] Jeri A Logemann, Hilda B Fisher, Benjamin Boshes, and E Richard Blonsky. Frequency and cooccurrence of vocal tract dysfunctions in the speech of a large sample of parkinson patients. *Journal of Speech and hearing Disorders*, 43(1):47–57, 1978.

[65] Brian Harel, Michael Cannizzaro, and Peter J Snyder. Variability in fundamental frequency during speech in prodromal and incipient parkinson's disease: A longitudinal case study. *Brain and cognition*, 56(1):24–29, 2004.

[66] Alexander M Goberman, Michael Blomgren, and Erika Metzger. Characteristics of speech disfluency in parkinson disease. *Journal of Neurolinguistics*, 23(5):470–478, 2010.

[67] WJ Barry and M Pützer. Saarbrucken voice database. *Institute of Phonetics, Universität des Saarlandes, http://www. stimmdatenbank. coli. uni-saarland. de*, 2007.

[68] Kay Elemetrics. Disordered voice database. *Model*, 4337, 1994.

[69] Tamer A Mesallam, Mohamed Farahat, Khalid H Malki, Mansour Alsulaiman, Zulfiqar Ali, Ahmed Al-nasheri, and Ghulam Muhammad. Development of the arabic voice pathology database and its evaluation by using speech features and machine learning algorithms. *Journal of healthcare engineering*, 2017, 2017.

[70] Ghulam Muhammad, Mansour Alsulaiman, Zulfiqar Ali, Tamer A Mesallam, Mohamed Farahat, Khalid H Malki, Ahmed Al-nasheri, and Mohamed A Bencherif. Voice pathology detection using interlaced derivative pattern on glottal source excitation. *Biomedical signal processing and control*, 31:156–164, 2017.

[71] Fabien Ringeval, Julie Demouy, György Szaszák, Mohamed Chetouani, Laurence Robel, Jean Xavier, David Cohen, and Monique Plaza. Automatic intonation recognition for the prosodic assessment of language-impaired children. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(5):1328–1342, 2010.

[72] Juan Rafael Orozco-Arroyave, Julián David Arias-Londoño, Jesus Francisco Vargas Bonilla, María Claudia Gonzalez-Rátiva, and Elmar Nöth. New spanish

speech corpus database for the analysis of people suffering from parkinson's disease. In *In Proc. Of the International Confer- ence on Language Resources and Evaluation (lrec)*, pages 342–347, Reykjavik, Iceland, 2014.

[73] Ugo Cesari, Giuseppe De Pietro, Elio Marciano, Ciro Niri, Giovanna Sannino, and Laura Verde. A new database of healthy and pathological voices. *Computers & Electrical Engineering*, 68:310–321, 2018.

[74] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

[75] R Viswanathan, P Khojasteh, B Aliahmad, SP Arjunan, S Ragnav, P Kempster, Kitty Wong, Jennifer Nagao, and DK Kumar. Efficiency of voice features based on consonant for detection of parkinson's disease. In *2018 IEEE Life Sciences Conference (LSC)*, pages 49–52. IEEE, 2018.

[76] J. Lee, S. Kim, and H. Kang. Detecting pathological speech using contour modeling of harmonic-to-noise ratio. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5969–5973, May 2014.

[77] J. Moon and S. Kim. An approach on a combination of higher-order statistics and higher-order differential energy operator for detecting pathological voice with machine learning. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 46–51, Oct 2018.

[78] Smitha, S. Shetty, S. Hegde, and T. Dodderi. Classification of healthy and pathological voices using mfcc and ann. In *2018 Second International Conference on Advances in Electronics, Computers and Communications (ICAECC)*, pages 1–5, Feb 2018.

[79] Shih-Hau Fang, Yu Tsao, Min-Jing Hsiao, Ji-Ying Chen, Ying-Hui Lai, Feng-Chuan Lin, and Chi-Te Wang. Detection of pathological voice using cepstrum vectors: A deep learning approach. *Journal of Voice*, 2018.

[80] S. E. Shia and T. Jayasree. Detection of pathological voices using discrete wavelet transform and artificial neural networks. In *2017 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, pages 1–6, March 2017.

[81] Musaed Alhussein and Ghulam Muhammad. Voice pathology detection using deep learning on mobile healthcare framework. *IEEE Access*, 6:41034–41041, 2018.

[82] Nam Trinh and Darragh O'Brien. Pathological speech classification using a convolutional neural network. In *IMVIP 2019: Irish Machine Vision & Image Processing*, Technological University Dublin, Dublin, Ireland, August 2019.

[83] Okko Räsänen and Jouni Pohjalainen. Random subset feature selection in automatic recognition of developmental disorders, affective states, and level of conflict from speech. In *Interspeech*, pages 210–214, 2013.

[84] Meysam Asgari, Alireza Bayestehtashk, and Izhak Shafran. Robust and accurate features for detecting and diagnosing autism spectrum disorders. In *Interspeech*, pages 191–194, 2013.

[85] Laura Verde, Giuseppe De Pietro, Mubarak Alrashoud, Ahmed Ghoneim, Khaled N Al-Mutib, and Giovanna Sannino. Leveraging artificial intelligence to improve voice disorder identification through the use of a reliable mobile app. *IEEE Access*, 7:124048–124054, 2019.

[86] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[87] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[88] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[89] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[90] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[91] Lu Lu, Yeonjong Shin, Yanhui Su, and George Em Karniadakis. Dying relu and initialization: Theory and numerical examples. *arXiv preprint arXiv:1903.06733*, 2019.

[92] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[93] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[94] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.

[95] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[96] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[97] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

[98] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[99] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[100] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

[101] Augustus Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016.

[102] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363. PMLR, 2019.

[103] Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

[104] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pages 4–11, 2014.

[105] Yasi Wang, Hongxun Yao, and Sicheng Zhao. Auto-encoder based dimensionality reduction. *Neurocomputing*, 184:232–242, 2016.

[106] Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*, 2017.

[107] Shahin Amiriparian, Michael Freitag, Nicholas Cummins, and Björn Schuller. Sequence to sequence autoencoders for unsupervised representation learning from audio. In *Proc. of the DCASE 2017 Workshop*, 2017.

[108] Hoo-Chang Shin, Matthew R Orton, David J Collins, Simon J Doran, and Martin O Leach. Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4d patient data. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1930–1943, 2012.

[109] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

[110] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[111] Xianxu Hou, Linlin Shen, Ke Sun, and Guoping Qiu. Deep feature consistent variational autoencoder. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1133–1141. IEEE, 2017.

[112] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. *arXiv preprint arXiv:1703.01925*, 2017.

[113] Dhruv Khattar, Jaipal Singh Goud, Manish Gupta, and Vasudeva Varma. Mvae: Multimodal variational autoencoder for fake news detection. In *The World Wide Web Conference*, pages 2915–2921, 2019.

[114] Xiaopeng Li and James She. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 305–314, 2017.

[115] Geoffrey E Hinton, Terrence Joseph Sejnowski, Tomaso A Poggio, et al. *Unsupervised learning: foundations of neural computation*. MIT press, 1999.

[116] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[117] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005.

[118] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 2013.

[119] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020.

[120] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[121] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.

[122] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[123] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[124] Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 151–161, 2011.

[125] Shuangfei Zhai and Zhongfei Zhang. Semisupervised autoencoder for sentiment analysis. *arXiv preprint arXiv:1512.04466*, 2015.

[126] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*, 2015.

[127] Abhishek Kumar, Prasanna Sattigeri, and Tom Fletcher. Semi-supervised learning with gans: Manifold invariance with improved inference. In *Advances in Neural Information Processing Systems*, pages 5534–5544, 2017.

[128] Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Russ R Salakhutdinov. Good semi-supervised learning that requires a bad gan. In *Advances in neural information processing systems*, pages 6510–6520, 2017.

[129] Jinhao Dong and Tong Lin. Margingan: Adversarial training in semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 10440–10449, 2019.

[130] Jun Deng, Nicholas Cummins, Maximilian Schmitt, Kun Qian, Fabien Ringeval, and Björn Schuller. Speech-based diagnosis of autism spectrum condition by

generative adversarial network representations. In *Proceedings of the 2017 International Conference on Digital Health*, pages 53–57. ACM, 2017.

[131] Nam Trinh and Darragh O'Brien. Semi-supervised learning with generative adversarial networks for pathological speech classification. In *31st Irish Signals and Systems Conference (ISSC2020)*, Letterkenny, Ireland (Virtual), June 2020.

[132] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[133] Mark A Friedl and Carla E Brodley. Decision tree classification of land cover from remotely sensed data. *Remote sensing of environment*, 61(3):399–409, 1997.

[134] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[135] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[136] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification, 2003.

[137] Thomas Murry and E Thomas Doherty. Selected acoustic characteristics of pathologic and normal speakers. *Journal of Speech, Language, and Hearing Research*, 23(2):361–369, 1980.

[138] Vijay Parsa and Donald G Jamieson. Acoustic discrimination of pathological voice. *Journal of Speech, Language, and Hearing Research*, 2001.

[139] Mazin Abed Mohammed, Karrar Hameed Abdulkareem, Salama A Mostafa, Mohd Khanapi Abd Ghani, Mashael S Maashi, Begonya Garcia-Zapirain, Ibon

Oleagordia, Hosam Alhakami, and Fahad Taha AL-Dhief. Voice pathology detection and classification using convolutional neural network model. *Applied Sciences*, 10(11):3723, 2020.

[140] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[141] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[142] Nitesh V Chawla and Grigoris Karakoulas. Learning from labeled and unlabeled data: An empirical study across techniques and domains. *Journal of Artificial Intelligence Research*, 23:331–366, 2005.

[143] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[144] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Towards deeper understanding of variational autoencoding models. *arXiv preprint arXiv:1702.08658*, 2017.

[145] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[146] M Ehsan Abbasnejad, Anthony Dick, and Anton van den Hengel. Infinite variational autoencoder for semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5888–5897, 2017.

[147] Shichen Cao, Jingjing Li, Kenric P Nelson, and Mark A Kon. Coupled vae: Improved accuracy and robustness of a variational autoencoder. *arXiv preprint arXiv:1906.00536*, 2019.

[148] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[149] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[150] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

[151] Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv preprint arXiv:1602.03616*, 2016.

[152] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.