PATHOLOGICAL VOICE DETECTION BASED ON MULTI-SCALE CONVOLUTIONAL NEURAL NETWORK

WONGSATHON PATHONSUWAN

A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy in Telecommunication
and Computer Engineering
Suranaree University of Technology
Academic Year 2023

การแยกแยะเสียงพยาธิวิทยาบนพื้นฐานโครงข่ายประสาทคอนโวลูชันแบบหลายมาตราส่วน

นายวงศธร ภาธรสุวรรณ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาปรัชญาดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคมและคอมพิวเตอร์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2566

PATHOLOGICAL VOICE DETECTION BASED ON MULTI-SCALE CONVOLUTIONAL NEURAL NETWORK

Suranaree University of Technology has approved this thesis submitted in partial fulfillment of the requirements for The Degree of Doctor of Philosophy.

Thesis Examining Committee

.....

(Assoc. Prof. Dr. Chanchai Thaijiam)
Chairperson

.....

(Assoc. Prof. Dr. Peerapong Uthansakul)
Member (Thesis Advisor)

.....

(Asst. Prof. Dr. Khomdet Phapatanaburi)
Member (Thesis Co-Advisor)

.....

(Assoc. Prof. Dr. Piyaporn Mesawad)
Member

.....

(Assoc. Prof. Dr. Monthippa Uthansakul)
Member

.....

(Dr. Dheerasak Anantakul)
Member

.....

(Assoc. Prof. Dr. Yupaporn Ruksakulpiwat)
Vice Rector for Academic Affairs
and Quality Assurance

.....

(Assoc. Prof. Dr. Pornsiri Jongkol)
Dean of Institute of Engineering

วงศธร ภาธรสุวรรณ : การแยกแยะเสียงพยาธิวิทยาบนพื้นฐานโครงข่ายประสาทคอนโวลูชันแบบหลายมาตราส่วน (PATHOLOGICAL VOICE DETECTION BASED ON MULTI-SCALE CONVOLUTIONAL NEURAL NETWORK)
อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร.พีระพงษ์ อุฑารสกุล และผู้ช่วยศาสตราจารย์ ดร.คมเดช ภาพัฒนบุรี, 111 หน้า

คำสำคัญ: การตรวจจับเสียงพยาธิวิทยา/โครงข่ายประสาทคอนโวลูชัน/ปัญญาประดิษฐ์

วิทยานิพนธ์นี้นำเสนอวิธีการใหม่ในการตรวจจับเสียงทางพยาธิวิทยาโดยใช้สถาปัตยกรรมโครงข่ายประสาทเทียมคอนโวลูชันแบบหลายมาตราส่วน (Multi-Scale Convolutional Neural Network: MSConvNet) วัตถุประสงค์ของการศึกษานี้คือการระบุความผิดปกติของเสียงจากข้อมูลเสียงที่ไม่ผ่านกระบวนการใด ๆ สถาปัตยกรรมใหม่นี้ถูกเรียกว่า RS-MSConvNet ถูกออกแบบมาเพื่อวินิจฉัยความผิดปกติของเสียงจากข้อมูลเสียงดิบ โมเดลนี้ใช้บล็อกคอนโวลูชันหลายมาตราส่วนเชื่อมต่อกับชั้นเชื่อมโยงสมบูรณ์ (Fully Connected Layer: FC) สำหรับการจำแนกประเภทโดยมุ่งหวังที่จะแยกแยะความแตกต่างของเสียงที่มีความผิดปกติและเสียงที่ปกติ

นอกจากนี้ วิทยานิพนธ์ฉบับนี้ยังนำเสนอ RS-MSConvNet-SVM ซึ่งเป็นโมเดลการเรียนรู้แบบผสมผสานระหว่างการเรียนรู้เชิงลึกและการเรียนรู้ของเครื่อง ซึ่งรวมความสามารถในการสกัดคุณลักษณะของ RS-MSConvNet กับประสิทธิภาพการจำแนกประเภทของ Support Vector Machine (SVM) เพื่อเพิ่มความแม่นยำในการระบุความผิดปกติของเสียง อีกทั้งยังใช้กลไกการคัดเลือกคุณลักษณะอย่าง Particle Swarm Optimization (PSO) ซึ่งเป็นวิธีการเพิ่มประสิทธิภาพ ผลการทดลองกับฐานข้อมูล TORGO ซึ่งประกอบด้วยตัวอย่างเสียงที่มีสุขภาพดีและเสียงที่ผิดปกติ ผลการทดสอบพบว่า RS-MSConvNet, RS-MSConvNet-SVM และ RS-MSConvNet-SVM ที่ใช้ PSO สามารถบรรลุความแม่นยำได้ 86.46%, 87.61% และ 88.09% ตามลำดับ ซึ่งผลลัพธ์แสดงให้เห็นว่าวิธีที่นำเสนอมีประโยชน์สำหรับการตรวจจับเสียงทางพยาธิวิทยา

สาขาวิชา วิศวกรรมโทรคมนาคม
ปีการศึกษา 2566

ลายมือชื่อนักศึกษา .....
ลายมือชื่ออาจารย์ที่ปรึกษา .....
ลายมือชื่ออาจารย์ที่ปรึกษาร่วม .....

WONGSATHON PATHONSUWAN: PATHOLOGICAL VOICE DETECTION BASED ON MULTI-SCALE CONVOLUTIONAL NEURAL NETWORK. THESIS ADVISOR: ASSOC. PROF. PEERAPONG UTHANSAKUL, AND ASST. PROF. KHOMDET PHAPATANABURI, Ph.D. 111 PP.

Keyword: PATHOLOGICAL VOICE DETECTION/END-TO-END ARCHITECTURE /MULTI-SCALE CONVOLUTION/SPATIALTEMPORAL FEATURE/HYBRID MODEL

This thesis proposes a new method for identifying pathological voice patterns by utilizing Multi-Scale Convolutional Neural Network (MSConvNet) architectures. The aim of the study is to detect abnormal voice characteristics from unprocessed speech data. A new architecture, namely RS-MSConvNet, has been designed to detect abnormal voice from raw speech. This model uses a multi-scale convolution block, a spatiotemporal feature block, and a fully connected layer for classification, with the goal of capturing differences between abnormal voice and normal voice.

Furthermore, the thesis proposes the RS-MSConvNet-SVM, a hybrid model that combines the feature extraction capabilities of RS-MSConvNet with the classification power of Support Vector Machine (SVM) to improve the accuracy of speech pathology identification. In addition, it utilizes a feature selection mechanism that employs Particle Swarm Optimization (PSO), a computational technique that enhances performance. Thorough experimentation with the TORGO database, which includes both normal and abnormal speech samples, revealed that the RS-MSConvNet, RS-MSConvNet-SVM, and RS-MSConvNet-SVM with PSO achieved remarkable accuracies of 86.46%, 87.61%, and 88.09%, respectively. The outcomes show that our proposed methods are useful for pathological voice detection.

School of Telecommunication Engineering Student's Signature .....

Academic Year 2023 Advisor's Signature .....

Co-Advisor's Signature .....

ACKNOWLEDGEMENT

The author wishes to acknowledge the funding support from Suranaree University of Technology (SUT).

I would like to express my sincere thanks to my thesis advisor, Assoc. Prof. Dr. Peerapong Uthansakul and Asst. Prof. Dr. Khomdet Phapatanaburi for his consistent supervision and thoughtful comments on several drafts and advice towards the completion of this study.

My thanks go to Assoc. Prof. Dr. Piyaporn Mesawad, Assoc. Prof. Dr. Chanchai Thaijiam, Assoc. Prof. Dr. Monthippa Uthansakul, and Dr. Dheerasak Anantakul for their valuable suggestions and guidance given as examination committees.

The author is also grateful to all faculty and staff members of the School of Telecommunication Engineering and colleagues for their help and assistance throughout the period of this work.

Finally, I would also like to express my deep sense of gratitude to the scholarship program, my parents and my advisor for their support and encouragement throughout the course of this study at the Suranaree University of Technology.

Wongsathon Pathonsuwan

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES (Continued)

Figure **Page**

LIST OF ABBREVIATIONS

ACC	=	Accuracy
ANN	=	Artificial Neural Network
ASR	=	Automatic Speech Recognition
BN	=	Batch Normalization
BP	=	Back Propagation
CNN or Conv	=	Convolutional Neural Network
DNN	=	Deep Neural Network
FC	=	Fully Connected
FN	=	False Negative
FP	=	False Positive
FP	=	Forward Propagation
GA	=	Genetic Algorithms
GCI	=	Glottal Closure Instant
GIF	=	Glottal Inverse Filtering
GMM	=	Gaussian Mixture Model
GPU	=	Graphics Processing Unit
KLD	=	Kullback Leibler Divergence
LDA	=	Linear Discriminant Analysis
LPC	=	Linear Prediction Coefficients
LPCC	=	Linear Predictive Cepstral Coefficients
LR	=	Learning Rates
LSTM	=	Long Short-Term Memory
MDVP	=	Multi-Dimensional Voice Program
MFCCs	=	Mel-Frequency Cepstral Coefficients
MLP	=	Multilayer Perceptron
MNIST	=	Modified National Institute of Standards and Technology
MSConvNet	=	Multi-Scale Convolution Neural Network

LIST OF ABBREVIATIONS (Continued)

MSE	=	Mean Squared Error
NLP	=	Natural Language Processing
openSMILE	=	open-source Speech and Music Interpretation by Large-space Extraction
PCEN	=	Per Channel Energy Normalization
PD	=	Parkinson's Disease
PSO	=	Particle Swarm Optimization
QCP	=	Quasi-Closed Phase
RBF	=	Radial Basis Function
RS	=	Raw Speech
SGD	=	Stochastic Gradient Descent
SVD	=	Saarbruecken Voice Database
SVM	=	Support Vector Machines
TN	=	True Negative
TNR	=	True Negative Rate
TP	=	True Positive
TPR	=	True Positive Rate
t-SNE	=	t-Distributed Stochastic Neighbor Embedding
TPU	=	Tensor Processing Unit
WLP	=	Weighted Linear Prediction

CHAPTER I
INTRODUCTION

1.1 Background and problem statement

Pathological voice detection is a vital technique used in voice healthcare systems, including voice clinics and telemonitoring applications, to distinguish between unhealthy or disordered voices and healthy voices. Voice clinics are specialized facilities where individuals with voice disorders receive diagnosis and treatment, while telemonitoring applications enable remote monitoring of patients' vocal health. By analyzing provided utterance signals and detecting changes in speech patterns, pathological voice detection serves as a valuable diagnostic tool for identifying the onset of disabling physical symptoms. The obtained results are utilized to screen patients at risk of specific diseases, enabling early intervention and treatment. Additionally, it plays a crucial role as a pre-processing step in other applications, such as dysphonic voice assessment for automatic speaker recognition. Dysphonia, characterized by abnormal vocal production, quality, or pitch, can be evaluated to determine the severity and type of voice disorder, aiding in appropriate treatment planning. Moreover, the technique is relevant in dysarthric speech recognition, assisting individuals with dysarthria, a motor speech disorder, to communicate more effectively by converting their speech into text or other forms of communication. Pathological voice detection falls within the domain of pattern recognition tasks in the field of biomedical and health informatics, where vocal features and patterns are analyzed to develop algorithms accurately distinguishing between healthy and pathological voices. This interdisciplinary approach combines expertise from biomedical sciences, informatics, and signal processing to advance the understanding and detection of vocal pathologies, ultimately improving patient care and outcomes.

Pathological voice detection systems can be broadly classified into two categories: traditional pipeline systems and modern end-to-end systems. Traditional pipeline systems, employed in earlier studies, consisted of a front-end feature extraction stage and a back-end classifier. The front-end stage involved converting speech signals into parametric representations using handcrafted design features, while the back-end classifier learned feature representations for classifying pathological and healthy voices. In contrast, modern end-to-end systems leverage deep learning techniques and eliminate the need for manual feature extraction. These systems utilize deep learning-based classifiers trained on raw speech or spectrogram data to predict the target classes. A comprehensive survey covering both traditional pipeline systems and modern end-to-end systems provides an overview of existing methodologies, techniques, and algorithms. It evaluates their strengths, limitations, and performance in pathological voice detection, aiming to identify areas for further improvement. The survey explores the historical development of feature extraction methods in traditional systems and assesses the advancements made by modern end-to-end approaches, facilitating the understanding and progress of pathological voice detection research. By combining the strengths of both approaches, future advancements can enhance the accuracy and efficiency of pathological voice detection, ultimately improving the diagnosis and treatment of voice disorders.

The primary focus of pathological voice detection research has centered around the traditional pipeline approach, which involves developing customized feature extraction techniques and employing suitable classifiers. Numerous methods for feature extraction have been introduced to enhance the identification of pathological voice conditions. These include Mel-Frequency Cepstral Coefficients (MFCCs), Linear Predictive Cepstral Coefficients (LPCC), Linear Prediction Coefficients (LPC), and features derived from the Multi-Dimensional Voice Program (MDVP) (Mesallam et al., 2017). Additional features like harmonic-to-noise ratio, Jitter, Shimmer, Kullback-Leibler Divergence (KLD) histogram, and KLD higher amplitude suppression

spectrum have also been proposed (Sabir, Rouda, Khazri, Touri, & Moussetad, 2017; Vaiciukynas, Verikas, Gelzinis, Bacauskiene, Kons, Satt, & Hoory, 2014; Barreira & Ling, 2020). Explorations have been made into autocorrelation and entropy features across different frequency regions (Al-Nasheri, 2018). Furthermore, efforts have been made to integrate acoustic features with statistical function sets and combine frequency-domain and time-domain glottal feature sets through the open-source Speech and Music Interpretation by Large-space Extraction (openSMILE) set or glottal source set-based fusion features. Researchers have even investigated the fusion of the openSMILE set and Glottal source set-based features to leverage the strengths of different feature types (Narendra & Alku, 2020). Support Vector Machines (SVM) have gained popularity as the preferred choice of classifiers in pathological voice detection due to their promising performance (Arjmandi & Pooyan, 2012; Sellam & Jagadeesan, 2014; Ali, 2016; Fang et al., 2019). However, alternative classifiers like Artificial Neural Networks (ANN), Linear Discriminant Analysis (LDA), Gaussian Mixture Model (GMM), and decision trees have also been explored (Ritchings, McGillion, & Moore, 2002; Teixeira, Fernandes, & Alves, 2017; Gómez-Vilda et al., 2009; El Emary, Fezari, & Amara, 2014; Hemmerling et al., 2016). These classifiers have been employed in various traditional pipeline approaches to effectively differentiate between pathological and healthy speech. The success of pathological voice detection within the traditional pipeline approach heavily relies on the effectiveness of handcrafted design feature extraction. Achieving accurate detection outcomes necessitates expertise and domain knowledge in speech processing, underscoring the significance of understanding speech processing techniques and possessing feature engineering proficiency to develop high-performance pathological voice detection systems.

When it comes to pathological voice detection using end-to-end systems, previous studies (Harar et al., 2017; Doshi et al., 2021; Kourkounakis, Hajavi, & Etemad, 2021) have demonstrated that expert feature engineering is not necessarily due to the capabilities of deep learning models to be trained using either the raw speech signal

or its spectrum. For instance, in Narendra and Alku (2020). combinations of Convolutional Neural Network and Multilayer Perceptron (CNN-MLP) or Long Short-Term Memory Networks (CNN-LSTM) were proposed, utilizing the raw speech signal as input. The results showed that these models, CNN-MLP and CNN-LSTM, achieved promising outcomes in pathological voice detection. However, using the raw speech signal without any modifications proved to be less efficient when working with limited training data. To address this limitation and further enhance the performance of the end-to-end CNN-MLP and CNN-LSTM, researchers in Harar et al. (2017) introduced the use of glottal flow signals as an alternative to the raw speech signal. The results indicated that employing glottal flow signals in the end-to-end CNN-LSTM and CNN-LSTM models yielded better performance compared to the conventional approaches using raw speech signals. Despite the encouraging results obtained by the end-to-end CNN-MLP and CNN-LSTM models with either raw speech or glottal source signals, there remains an ongoing research challenge to design novel end-to-end models specifically tailored for pathological voice detection. This research area offers opportunities for developing innovative approaches that can further advance the field and enhance the accuracy and efficacy of pathological voice detection systems.

1.2 Thesis objectives

The objectives of this thesis are as follows:

1.2.1 To study deep neural networks in the domain of pathological voice detection, focusing on understanding the underlying processes and mechanisms involved.

1.2.2 To design and propose a new end-to-end deep neural network model for pathological voice detection that does not require expert knowledge in feature engineering.

1.3 Scope and limitation of the thesis

1.3.1 The experimental results are based on the TORGO dataset (Rudzicz, Namasivayam, & Wolff, 2012) to ensure reliability in the data used for training and evaluating the model.

1.3.2 This thesis aims to propose a novel end-to-end deep neural network model using raw speech for pathological voice detection.

1.3.3. The proposed method is compared to five existing systems in detecting pathological voices. It outperforms their performance, thus confirming the substantial effectiveness and usefulness of the proposed approach for pathological voice detection.

1.3.4. In order to enhance the performance of the proposed approach, a hybrid method by integrating the feature extraction ability of the proposed model and the classifier of SVM method is implemented to further improve its effectiveness.

1.3.5. In order to improve the performance of the proposed approach, a feature selection method is implemented as an additional step to further improve its effectiveness.

1.4 Contributions

1.4.1 Propose RS-MSConvNet, a novel end-to-end multi-scale convolution neural network model using raw speech for pathological voice detection.

1.4.2 Propose RS-MSConvNet-SVM, a novel hybrid detection model by integrating the feature extraction ability of the RS-MSConvNet model and the classifier of support vector machine.

1.4.3 Explore feature selection to improve the performance of the proposed RS-MSConvNet-SVM method.

1.5 Organization of the thesis

The remainder of this thesis is outlined as follows: Chapter II provides an extensive exploration of the underlying theoretical framework, establishing the fundamental principles associated with this research. It serves as the bedrock by introducing the central concepts and principles of the thesis. Notably, it introduces the innovative concept of optimization without feature extraction, which represents a distinctive approach to deep learning for speech disorder prediction. Moreover, this chapter critically examines relevant literature, delving into previous studies on speech disorder prediction, deep learning methodologies, and optimization techniques.

Chapter III concentrates on elucidating the methodology employed in designing deep learning models for pathological detection. The chapter places particular emphasis on end-to-end pathological detection techniques, which eliminate the necessity for specialized knowledge in feature extraction. It delves into the diverse methodologies and techniques used to train and construct these models, providing insights into their implementation and deployment.

Chapter IV presents the results obtained from the experimental analysis of the developed models. It begins with an introduction and explores various optimization aspects of the RS-ConvNet model, including the impact of speech length, learning rate, batch size, momentum dynamics, and decay rate. The chapter then examines the effects of fully connected (FC) layers in the RS-MSConvNet model and the visualization of features learned by RS-MSConvNet. It concludes with a thorough performance analysis of RS-MSConvNet models, and a summary of the key findings presented in this chapter.

Chapter V presents the outcomes of the developed models, accompanied by comprehensive discussions. The model results are subdivided into three subsections, enabling an in-depth analysis. The first subsection showcases the models' performance when using different lengths of input speech, offering valuable insights into the influence of input length on the model's accuracy. The second subsection compares

the models' performance when employing different layers, facilitating an examination of the impact of layer configurations on detection outcomes. Lastly, the third subsection conducts a comparative analysis of the models' performance against baseline systems, specifically utilizing the TORGO database as a benchmark.

Chapter VI serves as the concluding chapter of the thesis, encapsulating the key findings and presenting conclusions drawn from the research. Additionally, this chapter provides valuable suggestions and recommendations for future studies, identifying potential areas for further investigation and improvement within the field of pathological detection.

CHAPTER II

BACKGROUND THEORY

2.1 Introduction

This chapter gives a detail on the end-to-end pathological voice detection model. It stresses how important it is to find and analyze abnormal voice patterns by using a mix of advanced methods and basic ideas. This chapter also goes into great detail about the basic ideas that this thesis is based on, which are based on the idea of proposed model.

2.2 Feedforward Neural Networks

First, learn how a basic feedforward neural network learns because it was the first and most straightforward Artificial Neural Network (ANN) ever made. This will help it make sense of more complicated recurrent neural network architectures (Schmidhuber, 2015).

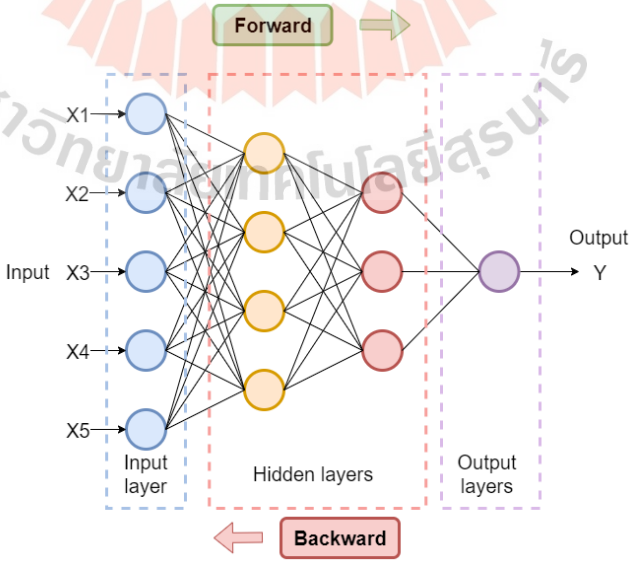


Figure 2.1 Feed-forward back-propagation mechanism in artificial neural network

2.1.2 Forward Propagation

The input data should be fed in the forward direction to generate output. The input data should not flow in reverse during output generation because it would form a loop, and the result could never be developed. These network configurations are feed-forward networks (Russell & Norvig, 2010). The feed-forward network helps in forward propagation.

$$z(x) = b + \sum_i w_i x_i = b + w^T x \tag{2.1}$$

Where b is the bias, w^T is the weight vector's transpose, and x is the input vector.

The activation or squash function permits neurons to turn their input into an output within a specific range. The activation function of neural networks must be non-linear to learn non-linear decision boundaries. Table. 2.1 Common activation functions in artificial neural networks

Table 2.1 Common activation functions in artificial neural networks.

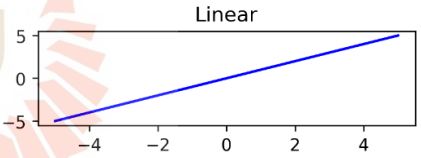
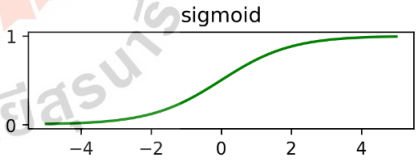
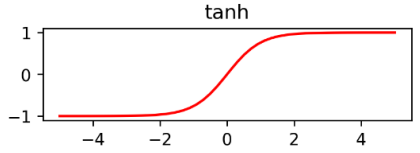
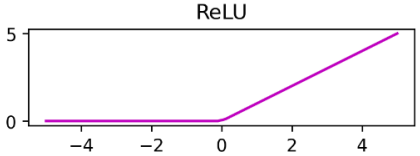
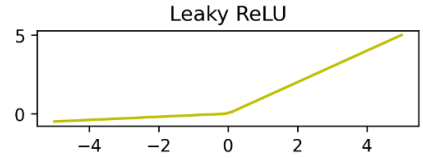
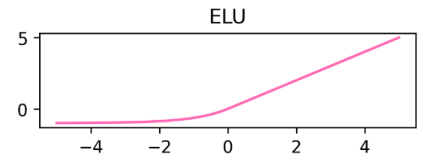
Name	Equation	Graph
Linear	$\sigma(z) = z$	
Sigmoid	$\sigma(z) = \frac{1}{1+e^{-z}}$	
Hyperbolic	$\sigma(z) = \tanh(z)$	
ReLu	$\sigma(z) = \begin{cases} z & \text{if } z > 0 \\ 0 & \text{if } z < 0 \end{cases}$	

Table 2.1 Common activation functions in artificial neural networks (continued).

Leaky ReLu	$\sigma(z)=\begin{cases} z & \text{if } z>0 \\ 0.1z & \text{if } z<0 \end{cases}$	
ELU	$\sigma(z)=\begin{cases} z & \text{if } z>0 \\ \alpha\left(e^{z}-1\right) & \text{if } z<0 \end{cases}$	

2.2.2 Backward Propagation

The technique of minimizing the error between the predicted output $h_w(x)$ and the intended output y by modifying the weights using derivatives of a loss function $L(w)$ is known as backward propagation (sometimes referred to as back-propagation or BP) (Russell & Norvig,2010). Therefore, the process entails propagating the error from the output layer and iteratively propagating it back through the hidden levels. When employing the Mean Squared Error (MSE) as the loss function, the following equation holds for any weight W:

$$\frac{\partial}{\partial w}L(w)=\frac{\partial}{\partial w}\left|y-h_w(w)\right|^2=\frac{\partial}{\partial w}\sum_k\left(y_k-\sigma_k(z)\right)^2=\sum_k\frac{\partial}{\partial w}\left(y_k-\sigma_k(z)\right)^2 \quad (2.2)$$

Where k pertains to the nodes located in the output layer, the elements within the final summation comprise the gradient of the loss function for each k output item. This approach allows us to decompose the overarching challenge of learning m outputs. By aggregating gradients from each k output element, it can effectively compute the overall error gradient $\frac{\partial}{\partial w}L(w)$. It should be noted that both y and $h_w(x)$ are vector quantities. The back-propagation is distinct for neurons in the output layer and the concealed layers.

2.2.2.1 Back Propagation for neurons in the output layer

Since the parameters of the output layer directly affect the loss, it is relatively simple to compute the loss gradients for those units. It defines the loss for each output unit k as the difference between prediction h_k and target y_k. Applying the chain rule results in the loss gradient expression:

$$\frac{\partial}{\partial w_{jk}} L(w) = \left(\sigma_k(z_k) - y_k\right)\sigma_k'(z_k)\sigma_j(z_j) = \Delta_k \sigma_j(z_j) \tag{2.3}$$

where the modified error is defined as:

$$\Delta_k = \left(\sigma_k(z_k) - y_k\right)\sigma_k'(z_k) \tag{2.4}$$

The initial term represents the discrepancy between the activation $\sigma(z)$ of the network's output layer and the target value y for each element k. The derivative of the activation function of the output layer is referred to as the second term. The activation output of the j node in the hidden layer is denoted as σ_j.

Subsequently, equation 2.3 is employed in the delta rule to iteratively update weights using the learning rate η.

$$w_{jk} \leftarrow w_{jk} + \eta \frac{\partial}{\partial w_{jk}} L(w) \tag{2.5}$$

2.2.2.2 Back Propagation for neurons in the hidden layers

In the case of hidden layers, it applies a propagation rule that is similar. However, it incorporates a modified error term of the output, considering that hidden node j bears responsibility for a certain degree of the error Δ_k in each output node k.

$$\Delta_j = \sigma'(z_j)\sum_k \Delta_k w_{jk} \tag{2.6}$$

Consequently, the loss gradient expression for hidden node j is obtained.

$$\frac{\partial}{\partial w_{ji}} L(w) = \Delta_j \sigma_i(z_i) \tag{2.7}$$

And concludingly, the weight-update rule for hidden layers is identical to its version in the output layer (equation 2.5).

$$w_{ij} \leftarrow w_{ij} + \eta \frac{\partial}{\partial w_{ij}} L(w) \tag{2.8}$$

Concludingly, the Back Propagation algorithm is responsible for the learning process in neural networks. It involves the iterative computation of the error between the predicted output value and the target output value, followed by the adjustment of weights in a way consistent with gradient descent. A visual depiction can also elucidate this concept, as shown in Figure 2.2. In this graphical representation, the iterative process of approaching the global minimum of the loss function involves following the gradient direction with step sizes denoted as η.

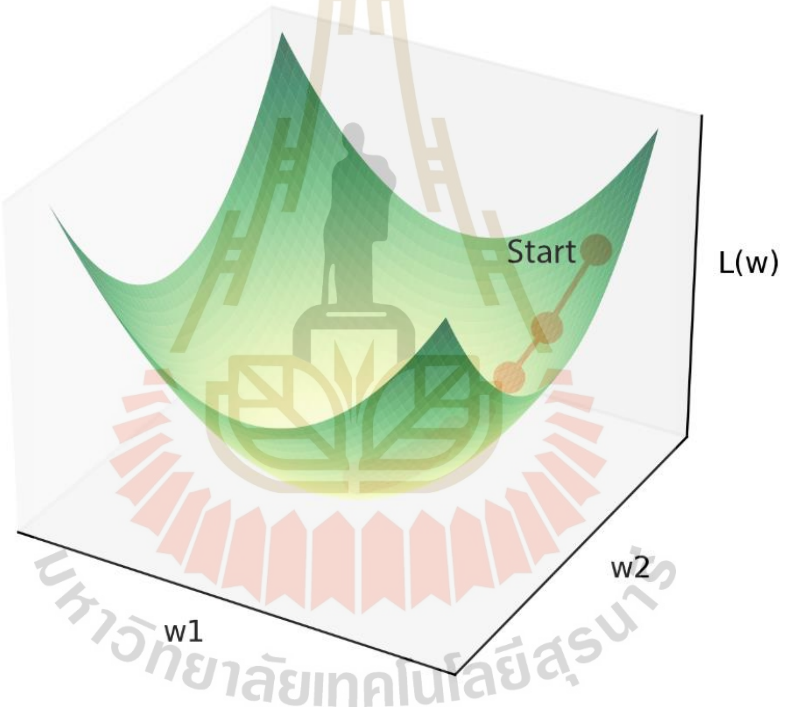


Figure 2.2 Gradient descent and the loss function $L(w)$

2.3 Convolutional Neural Network (CNN or Conv)

Convolutional Neural Network (CNN) are artificial intelligence that use multiple layers of neural networks to identify, recognize, and group objects. They can also find and separate objects in images. In reality, CNN or Conv is a well-known discriminative

deep learning architecture that can learn from the input object without requiring humans to extract features (Koushik, 2016; Bezdan & Džakula, 2019; Bouchard, 2022). This network is often used for visual identification, medical image analysis, image segmentation, Natural Language Processing (NLP), etc. Because it is designed to work with a wide range of 2D shapes, it works better than a standard network because it can automatically pick out essential parts of the input without help from a person.

In CNN, the layer comprises filters or kernels that process data. Initially given random weights, these kernels learn to extract features from input data through training, turning random integers into meaningful weights. CNN understands multi-channeled images, like RGB images with three color channels or single-channeled grayscale images, while traditional neural networks work with vector data. Applying these kernels to the input image and showing the results as N-dimensional metrics makes the feature map. Within a high-dimensional, implicit feature space, the kernel finds the inner products of all data pairs without knowing the coordinates of the data in that space. This “kernel trick” turns a linear model into a non-linear model to make CNN better at learning and pulling out complex features.

CNN needs a convolutional layer comprising different filters or kernels that change the input data. At first, these kernels were given random weights. It gradually changes based on training data to pull out features from the input, which are shown as N-dimensional metrics. During training, each kernel, which comprises a set of integers, learns how to pull out important features. This lets CNN work in a high-dimensional feature space without having to compute data coordinates directly. Instead, it figures out the inner products in the feature space. The kernel trick can be used on a linear model to turn it into a non-linear model. CNN's input format is set before convolution, which differs from traditional neural networks using vector format. CNN can handle images with multiple channels.

Examples of the convolution process are RGB images with three channels or grayscale images with one channel. Find patterns in a 4x4 grayscale image using a 2x2 kernel set up with random weights. Additionally, while the kernel moves across the image horizontally, the dot product of the input image and the kernel is being

calculated. Coordinate values are multiplied and added to get a single scalar value for this calculation. After each iteration, this process is repeated until the kernel can't move across the image anymore.

Transitioning from this practical example, it becomes evident that the underlying parameters greatly determine the effectiveness of such convolution operations. Parameters defining Convolution operations play a crucial role in shaping the outcome of these processes. They are the backbone that dictates how the convolution is executed, influencing everything from the precision of pattern detection to the speed and efficiency of the operation. Understanding these parameters is key to optimizing convolution processes for different applications and requirements.

Building further on the importance of these parameters, in the realm of CNN, they take on an even more significant role. In the realm of CNN, the convolutional layer serves as a fundamental building block, primarily utilized for feature extraction and spatial analysis in multidimensional data. The efficacy and efficiency of these layers are largely governed by a set of parameters that define their operational behavior. These parameters not only influence the transformation of data within the layer but also have far-reaching implications on the overall network architecture, computational complexity, and the nature of features extracted. Understanding these parameters is pivotal for designing effective CNN models. The primary parameters that define the operations of a convolutional layer are (PyTorch,2023):

1) **Input dimensions** $(N, C_{in}, H_{in}, W_{in})$: The input to a convolutional layer is typically a four-dimensional tensor, characterized by the batch size (N), the number of input channels (C_{in}), and the spatial dimensions - height (H_{in}) and width (W_{in}). These dimensions represent the size and complexity of the input data.
2) **Kernel size**: The kernel, or filter, is a small matrix used to apply the convolution operation. The size of the kernel (height and width) determines the extent of the local area in the input to which the convolution is applied. Larger kernels

encompass more input units, capturing broader features, while smaller kernels focus on finer, localized features.

3) **Padding**: Padding involves adding extra pixels around the edge of the input tensor. This technique is used to control the spatial dimensions of the output tensor, allowing for adjustments in feature map size, and is critical for handling edge cases where the kernel overlaps the bounds of the input.

4) **Stride**: Stride defines the step size with which the kernel moves across the input tensor. A larger stride results in broader spatial sampling, leading to smaller output dimensions, whereas a smaller stride offers finer sampling, preserving more spatial information in the output.

5) **Dilation**: Dilation refers to the spacing between the elements within the kernel. This parameter allows the kernel to expand and cover a larger receptive field without increasing the number of parameters. Dilation introduces an additional level of flexibility in manipulating the field of view of the convolutional filters.

The output dimensions $(N, C_{out}, H_{out}, W_{out})$ of a convolutional layer are influenced by its configuration. In this case, (C_{out}) is determined by the number of filters. The height (H_{out}) and width (W_{out}) of the output are calculated as follows:

$$H_{out} = \left\lfloor \frac{H_{in} + 2 \times padding[0] - dilation[0] \times \left(kernelsize[0] - 1\right) - 1}{stride[0]} + 1 \right\rfloor \quad (2.9)$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2 \times padding[1] - dilation[1] \times \left(kernelsize[1] - 1\right) - 1}{stride[1]} + 1 \right\rfloor \quad (2.10)$$

Figure 2.3 shows the main calculations that were done at each stage. The kernel is shown in the smaller square (2x2), and the input picture is shown in the larger square (4x4). Then, a product is shown as a number multiplied by both. This sum is used as an input value for the output feature map (Koushik, 2016; Bezdan & Džakula, 2019; Bouchard, 2022).

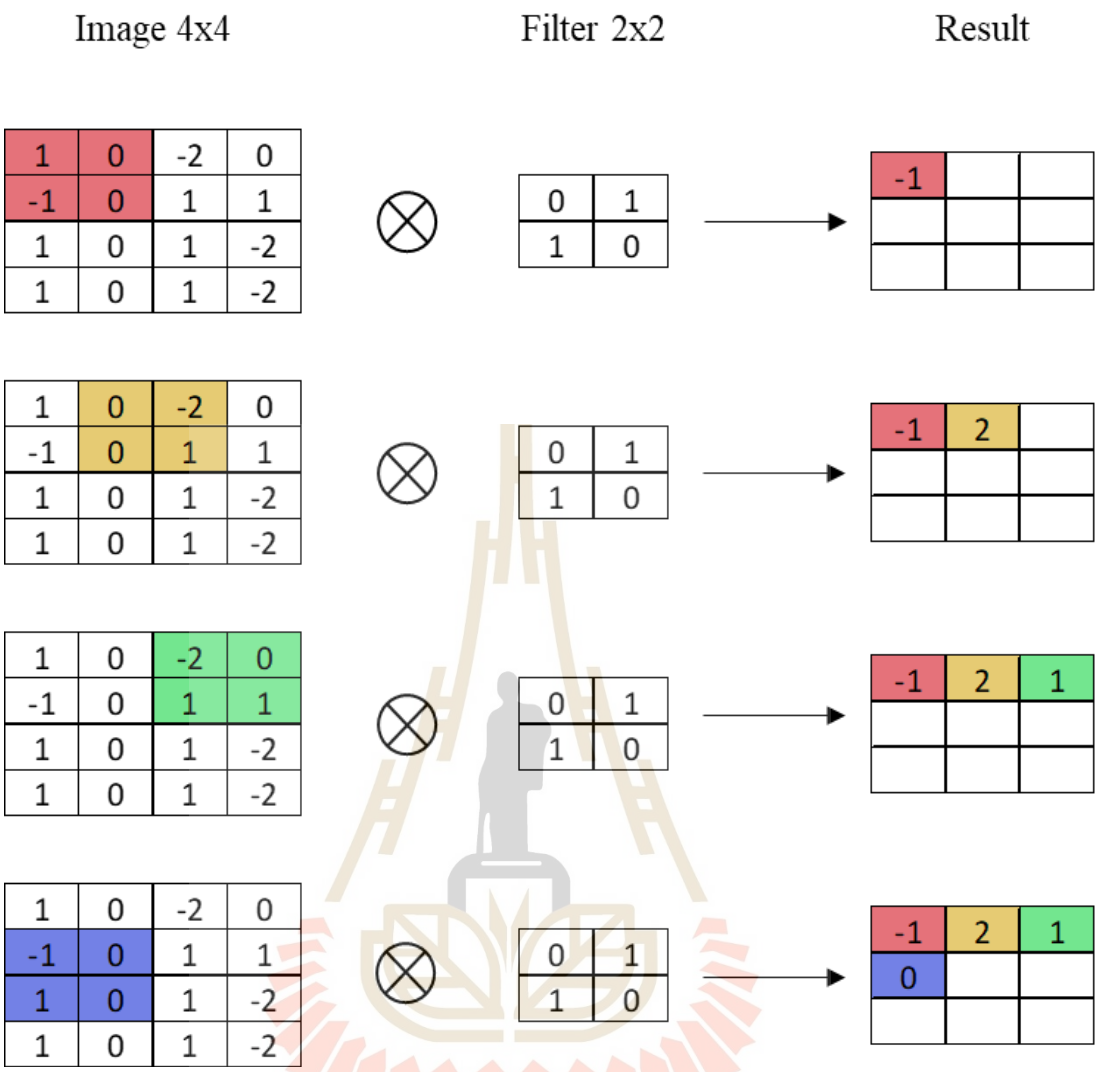


Figure 2.3 A picture that shows the main calculations

In the example mentioned before, the kernel works with a stride of 1, which sets the step size across the input image in both the vertical and horizontal directions. It is important to note that this example does not add padding to the input image. Remember that this method is flexible; a different stride value can be chosen based on specific needs. One of the best things about choosing a higher stride value is that it makes the feature map less multidimensional. This change can be significant for managing the feature map's size and complexity that the convolutional process creates.

However, padding dramatically affects the size of the picture's borders. It differs from the border side, which changes significantly over time. When the picture gets bigger, the feature map gets bigger, too. Each filter could stand for a feature. If the filter moves over an image and does not find a match, it does not work. CNN uses these steps to find the best object-description filters. Figure 2.4 shows how the matrix can be set up to see the edges of a picture. These matrices are also called filters because they work like the filters usually used in image processing.

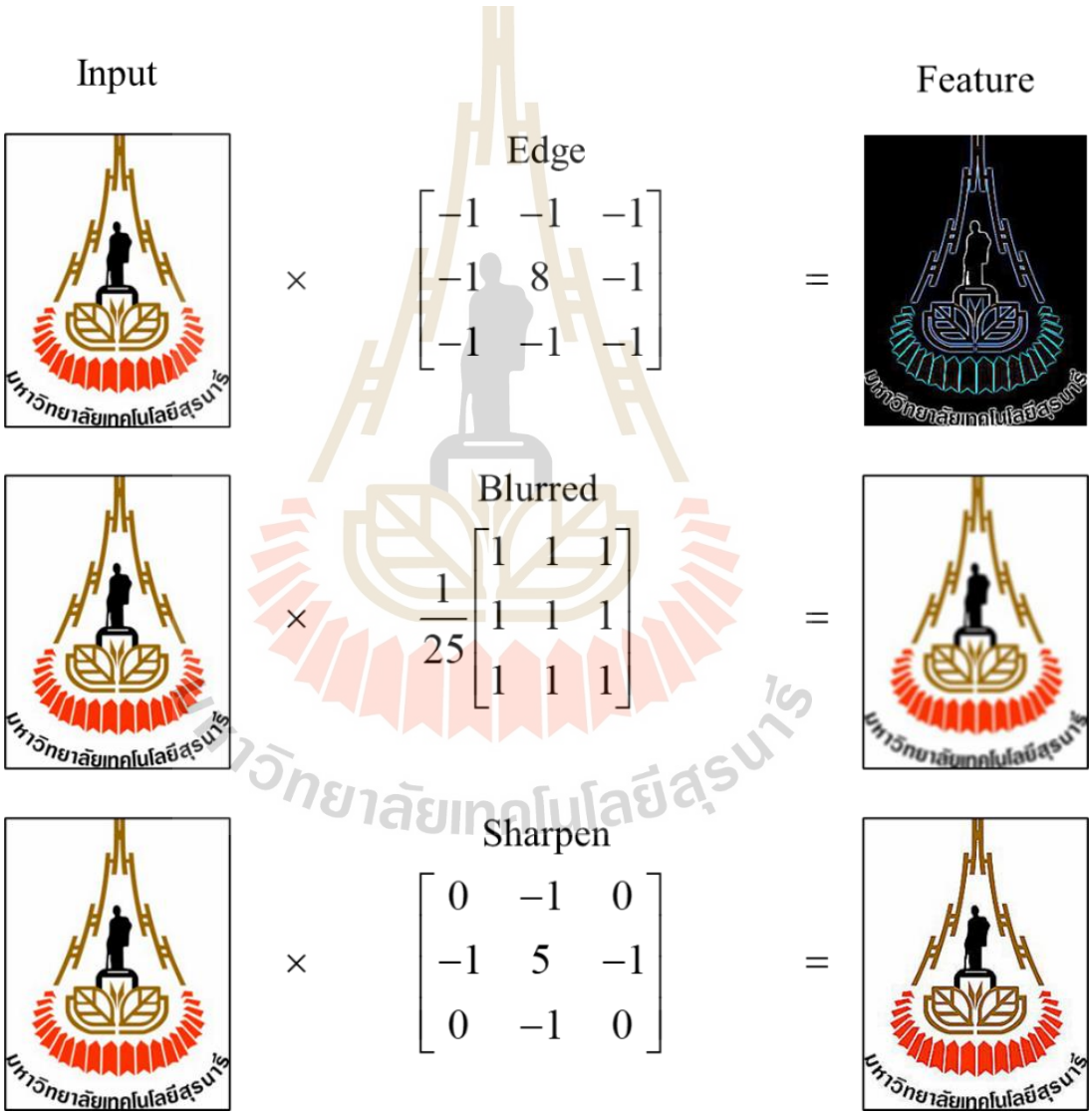


Figure 2.4 Effects of different convolution matrices

But at CNN, these filters are used before the form filters that are better for the job at hand and are used during training.

Stride: In fact, CNN gives it more choices that let it narrow down the settings even more while also lowering some of the harmful effects. The word for one of these is stride. From looking at the areas, the next-layer node overlaps with its neighbors in the situation described above. It can change the overlap by altering the stride. Figure 2.5 shows a one-of-a-kind 6x6 picture. The most significant output size it can get is 4 × 4 because the filter can only be moved by one node at a time. The outputs of the three left matrices, along with those of the three middle matrices and the three correct matrices, can be seen in Figure 2.5. With a walk and counting each step as two, it will be three times three. This means that the output's total size and the overlap amount will go down (Koushik, 2016; Dhillon & Verma, 2019; Edureka, 2022).

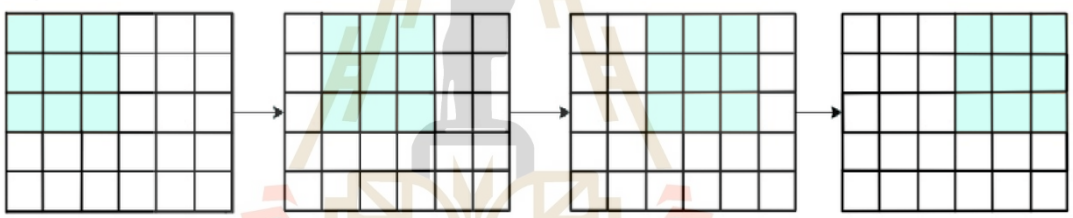


Figure 2.5 Stride 1, the filter windows move only one time for each connection.

Equation 2.11 shows a simple that leads to the output size, as seen in Figure 2.6. The size of the image ($D \times D$) and the size of the filter ($K \times K$) are used to figure this out.

$$output = 1 + \frac{D - K}{S} \tag{2.11}$$

where D is the input size, K is the filter size, and S is the stride size.

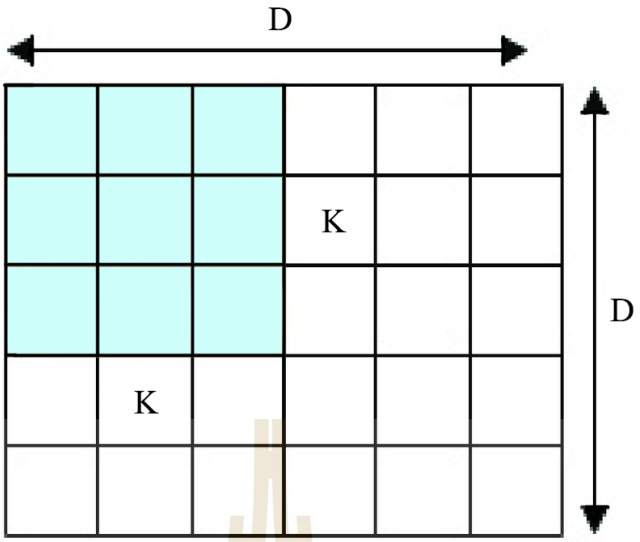


Figure 2.6 The effect of stride

Padding: The loss of detail at the edges of the image is one of the problems with the convolution step. It may not be able to see them because they are only picked up when the filter is moved. Using zero padding is a simple and helpful suggestion. When it uses zero padding, it can also change the output size.

For example, in Figure 2.6, the output will be 4 × 4, less than the input of 6 × 6. This is because D = 6, K = 3, and stride 1 were used. However, incorporating a single layer of zero-padding results in a 6 × 6 output identical to the original input. In this scenario, the actual dimension D becomes 9. This adjustment is reflected in the modified formula, which accounts for the scenario without padding, as detailed in Equation 2.12.

$$output = 1 + \frac{D + 2P - K}{S} \tag{2.12}$$

Where P is the number of layers of zero-padding (for example, P = 1 in Figure 2.7), this padding idea keeps the network output size from getting smaller as the depth goes up.

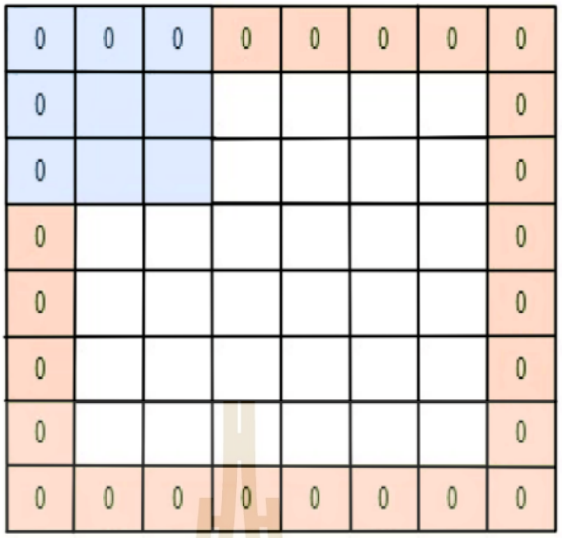


Figure 2.7 Zero-padding

2.4 Convolutional Neural Network for audio classification

In the expanding realm of deep learning applications, CNN have shown remarkable efficacy in the field of audio classification. This capability extends the versatility of CNN beyond visual data, enabling them to analyze and categorize complex audio patterns.

2.4.1 Understanding audio data for CNN

Audio data differs significantly from image data in its structure and representation. It is typically represented as a time series of amplitude values, often transformed into spectrograms or MFCCs for effective processing. These transformations convert audio into a 2D format (time vs frequency or cepstral coefficients), making it analogous to image data and thus suitable for CNN analysis.

2.4.2 Preprocessing and feature extraction

The first step in audio classification using CNN is preprocessing and feature extraction. This involves converting raw audio signals into a suitable format, such as spectrograms or MFCCs. These representations capture essential features like

frequency and time, allowing the network to identify patterns associated with different audio classes.

2.4.3 CNN architecture for audio data

The architecture of a CNN for audio classification is similar to that for image processing but tailored to the characteristics of audio data. The convolutional layers learn to recognize patterns in the frequency and time dimensions of the input features. Pooling layers reduce dimensionality and help in capturing invariant features. Fully connected layers towards the end of the network aid in classifying the audio into various categories based on the learned features.

2.4.4 CNN classification for pathological voice detection

CNN application for pathological voice detection represents a significant advancement in medical diagnostics, leveraging the power of deep learning to identify and classify voice disorders. This subsection delves into the specifics of using CNN for this purpose.

Understanding pathological voice and characteristics: Pathological voices are those affected by various disorders, such as nodules, polyps, or laryngeal diseases, which alter the typical characteristics of a person's voice. These alterations can manifest in various ways, such as pitch, volume, and quality changes. The challenge lies in accurately detecting these subtle changes, which may be imperceptible to the human ear but are critical for early diagnosis and treatment. This chapter gives a detail on the end-to-end pathological voice detection model.

Data collection and preprocessing: The first step for CNN-based pathological voice detection involves collecting a comprehensive dataset of healthy and pathological voice recordings. These recordings are then converted into a suitable format for analysis, like spectrograms or MFCCs, which effectively capture the unique attributes of each voice sample.

CNN architecture for pathological voice detection: The CNN architecture for this task is designed to extract and learn the intricate patterns associated with pathological voices. This involves:

1) Convolutional layers: These layers are adept at extracting local and global features from the spectrograms or MFCCs, focusing on characteristics that differentiate pathological voices from normal ones.
2) Pooling layers: They reduce the dimensionality of the data, enhancing the model's ability to generalize and focus on the most relevant features.
3) Fully Connected and output layers: The final stages of the network, where the learned features are used to classify the voice as either pathological or healthy.

Based on this conceptual framework, the architecture will show how convolutional, pooling, and fully connected layers are integrated to create a robust model that can identify pathological voices. This figure highlights the interplay among these layers as well as how they work together to improve the accuracy and dependability of the model. Figure 2.8 shows an example of CNN architecture for the detection of pathological voices.

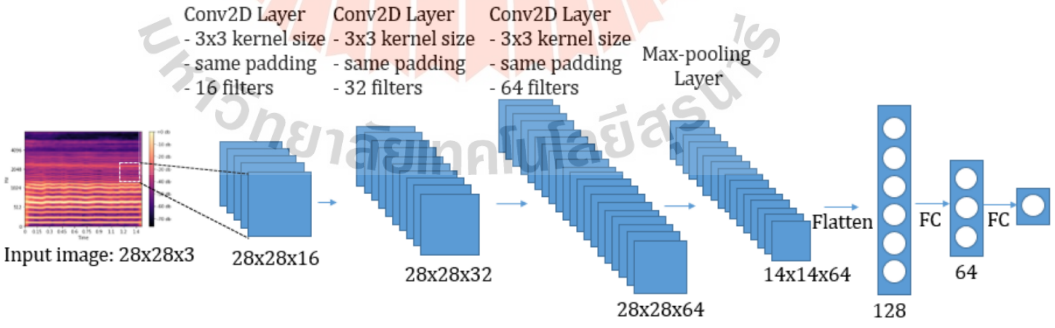


Figure 2.8 CNN architecture (Trinh & Darragh, 2019)

2.5 End-to-End CNN for pathological voice detection

The introduction of the end-to-end CNN tailored for pathological voice detection marks a transformative era in medical diagnostics. This pioneering CNN model revolutionizes the diagnostic approach by directly processing raw audio data. Figure 2.9 shows an example of the end-to-end CNN architecture for the detection of pathological voices. This novel method contrasts conventional models, which typically depend on extensive preprocessing. By directly handling raw data, this model streamlines the diagnostic process and substantially improves the precision and efficiency in identifying voice pathologies, offering a more direct and refined approach to diagnostics. Comparing general CNN and end-to-end CNN in detecting pathological voices

2.5.1 Core functionality

General CNN: These networks have been the backbone of visual data analysis, adept in tasks such as object recognition, image segmentation, and natural language processing. They shine in managing 2D image data, skillfully extracting vital features from both RGB and grayscale images.

End-to-end CNN for pathological voice detection: This specialized network is meticulously engineered for the complex realm of audio analysis, concentrating on detecting and categorizing pathological voice patterns. It diverges from general CNN's intrinsic ability to process unrefined audio data, seamlessly converting it into analyzable formats for in-depth examination.

2.5.2 Input data handling

General CNN: These networks typically engage with well-structured, multi-channeled image data, necessitating individualized processing of each channel, such as RGB, relying heavily on pre-processed and formatted data.

End-to-end CNN for pathological voice detection: Uniquely designed to manage raw, unstructured audio data directly, this network obviates the need for

conventional preprocessing practices like spectrogram or MFCCs conversion, typically indispensable in standard audio processing.

2.5.3 Architecture and feature extraction

General CNN: Composed of a diverse array of filters or kernels, each calibrated to identify and learn patterns in image data, thereby facilitating the classification of a wide range of objects within these images.

End-to-end CNN for pathological voice detection: Boasts an architecture meticulously crafted to align with the specificities of audio data. Its convolutional layers are fine-tuned to pinpoint unique voice patterns, mainly focusing on anomalies and irregularities that signal voice pathologies.

2.5.4 Layers and operations

General CNN: Employ a variety of convolutional layers featuring different filter sizes, padding, and stride configurations, enabling them to capture an extensive range of features, from basic to highly intricate.

End-to-end CNN for pathological voice detection: It utilizes convolutional, pooling, and fully connected layers but is focused on extracting features relevant to vocal quality and specific auditory patterns, differing markedly from visual feature extraction.

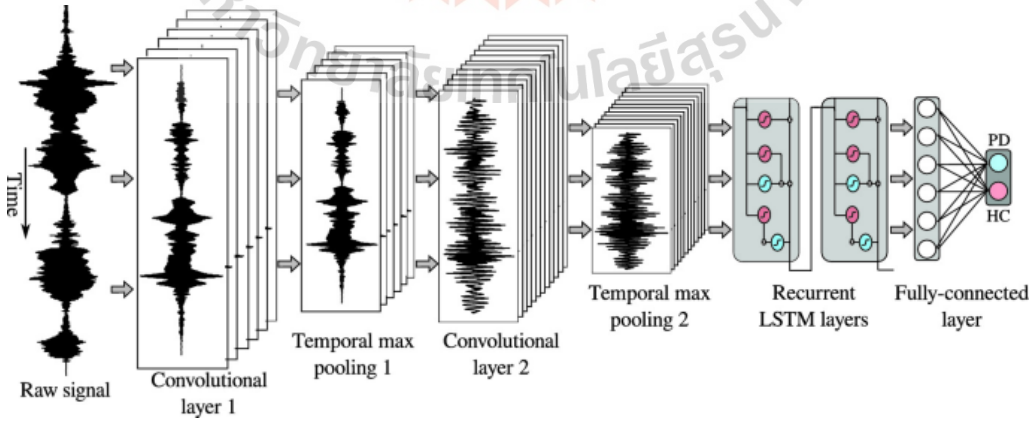


Figure 2.9 End-to-End CNN Architecture (Rios-Urrego et al., 2022)

2.6 End-to-End Multi-Scale Convolution Network for pathological voice detection

This section delves deeper into the model's detailed features. The multi-scale feature extraction, achieved through convolution operations at various scales, allows for the extraction of both detailed and extensive features, ranging from fine-scale detection of subtle voice irregularities to coarser scales assessing overall voice quality. The multi-scale analysis enhances sensitivity, enabling the detection of minute voice anomalies that single-scale analysis might overlook. This approach also facilitates a more holistic understanding of the voice sample through comprehensive feature mapping.

The advanced architecture and operations of the model are further explored, particularly its layer structure and data processing mechanics. The model comprises parallel convolutional pathways, each tailored to a specific scale, processing input data to extract relevant features. For example, one pathway might be adept at capturing the delicate nuances of voice tremors. These extracted features from different scales are then integrated using a mathematical fusion strategy, which can be expressed as:

$$F_{combined} = \alpha F_{fine-scale} + \beta F_{mid-scale} + \gamma F_{coarse-scale} \quad (2.13)$$

In the model's design, the weights α, β, γ are assigned to features from each scale, determining their contribution to the combined feature set. To ensure the precise capture of scale-appropriate features, the model utilizes optimized convolutional operations with customized kernel sizes for each scale. This attention to detail is evident in the varied kernel sizes employed: smaller kernels in fine-scale pathways focus on detailed aspects, while larger kernels in coarse-scale pathways are designed to capture broader patterns, as shown in Figure 2.10. This nuanced approach to feature extraction is vital for the model's accuracy and efficiency.

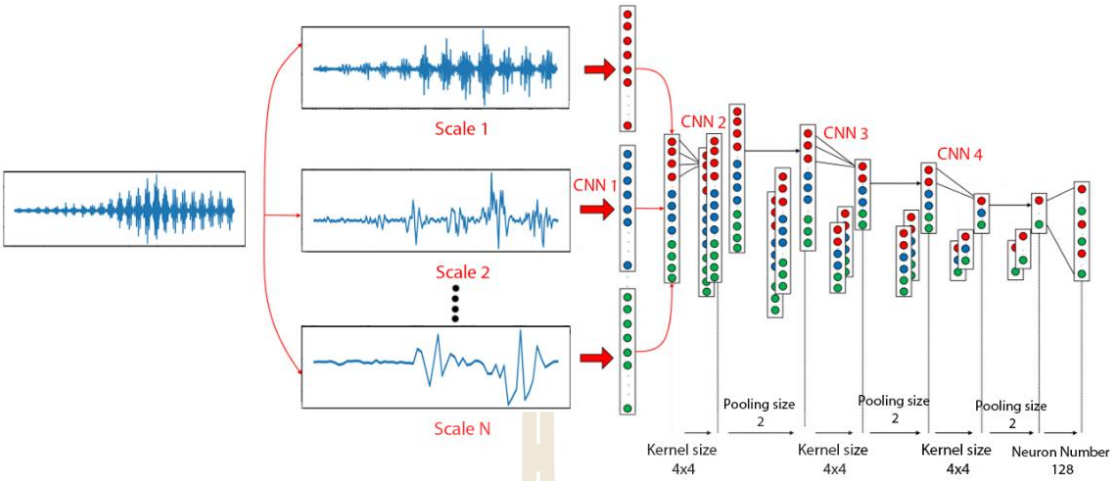


Figure 2.10 Multi-scale Convolutional Neural Network

Transitioning from these architectural specifics, it's important to address the overarching challenges the model faces, particularly in computational efficiency and interpretability. Overcoming these challenges involves a multifaceted approach. Strategies for computational management, such as network pruning and optimization, play a crucial role in reducing the network's complexity. This is done without compromising the model's ability to extract multi-scale features. In parallel, leveraging advanced computational hardware like Graphics Processing Units (GPUs) and cloud Tensor Processing Units (TPUs) is essential to manage the increased processing demands.

Moreover, the model's effectiveness is further enhanced by a robust training process. Utilizing expansive and diverse datasets is critical in improving the model's generalizability and robustness across various voice pathology scenarios. Finally, the model also prioritizes interpretability, especially in clinical settings. This is achieved through layer-wise feature visualization techniques, which allow clinicians to interpret the contributions of different scales to the final diagnosis. Such interpretability is crucial for practical application, bridging the gap between sophisticated machine learning techniques and real-world clinical utility.

2.7 Tuning a Neural Network

2.7.1 Pooling

Typically, there is a subsampling or pooling layer next to each convolutional layer. This combination lowers the resolution of the feature map, which makes the output less likely to shift or become distorted. (LeCun & Bengio, 1995). The following are the most commonly used pool pooling algorithms:

Max pooling: Max pooling chooses the pixels in the image that are brighter. It helps only to want to see the lighter pixels in an image, and the background is dark. Figure 2.11 shows that a pooling operation figures out the highest value for each window range sends that neuron out, and then moves it on to the next layer.

Average pooling: The average pooling method smooths out the image, and hence, the sharp features may not be identified when this pooling method is used. The average pooling is shown in Figure 2.12.

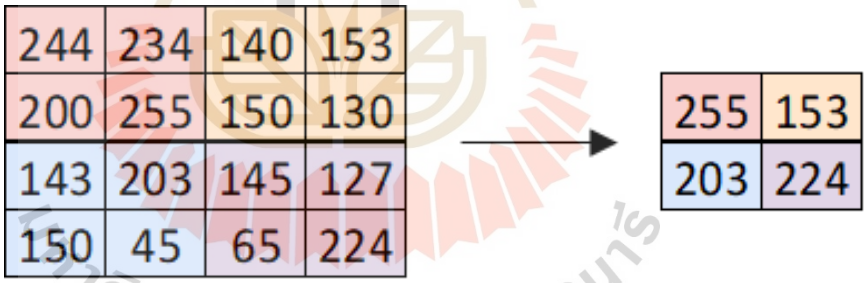


Figure 2.11 Max pooling

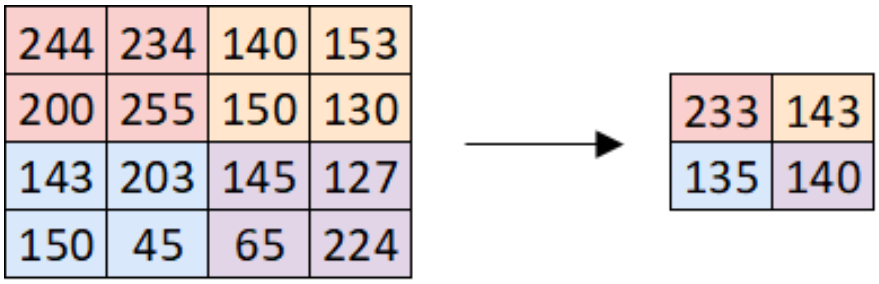


Figure 2.12 Average pooling

For example, Figure 2.13 shows a visual comparison between an original image of a purple flower and two processed versions of the same image using different pooling methods, which are techniques used in image processing to reduce the size of the image while retaining important features. The Original Image at the top is a detailed and clear photograph of the flower. To its bottom left, the image labelled “Average pooling” appears as a diminished rendition, with a noticeable decrease in resolution and a softened detail profile. This is indicative of the average pooling technique, where pixel values within a designated area are combined, assigning the mean value to the entire region, resulting in a more uniform but less detailed image. Conversely, in the bottom right, the “Max pooling” image, despite its lower resolution, conspicuously conserves more of the flower's intricate details and contours. This effect arises from the max pooling method, which selects the peak value from a specified pixel cluster to represent the whole area, thus maintaining more textural and edge clarity.

In conclusion, this set of images demonstrates the effects of two common pooling operations used in image processing and machine learning, particularly in CNN. Average pooling results in smoother but less detailed images, whereas max pooling preserves more detail at the cost of losing some smoothness. These techniques are critical in reducing computational load and extracting robust features for tasks such as image classification and pattern recognition.

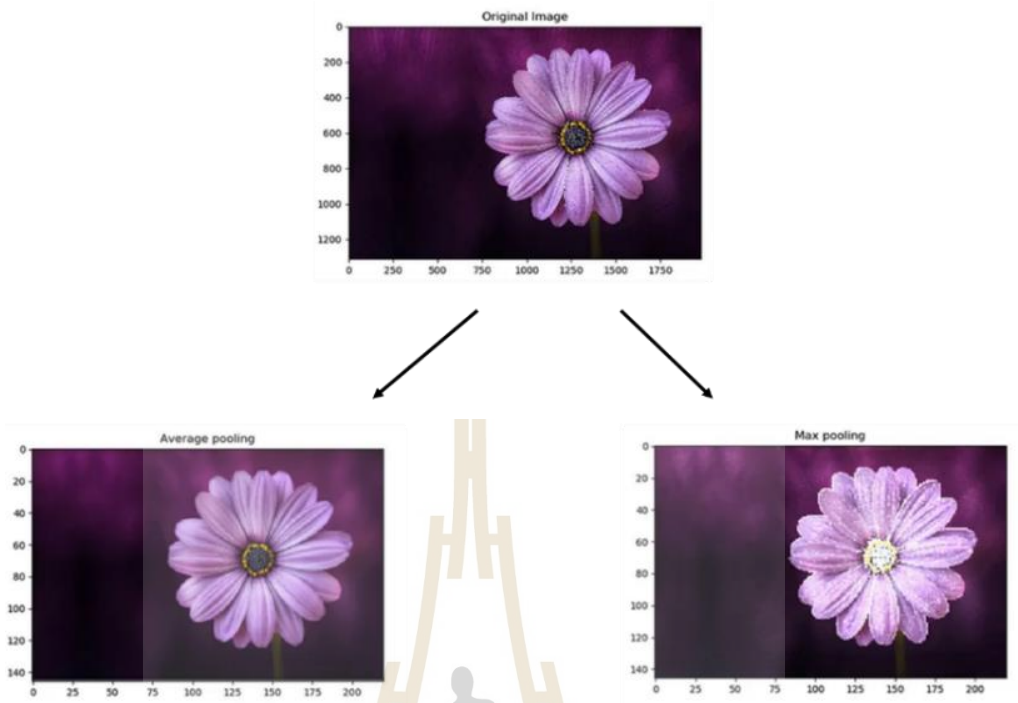


Figure 2.13 Average pooling and Max pooling (Dhuma, 2019)

2.7.2 Dropout

In 2014, Srivastava et al. (2014) introduced a technique called Dropout to prevent overfitting by randomly setting input units and dropping their related connections in the network during training. This means that the neurons will adapt to optimal weights that are less dependent on the weights and performance of other neurons. Dropout as shown in Figure 2.14

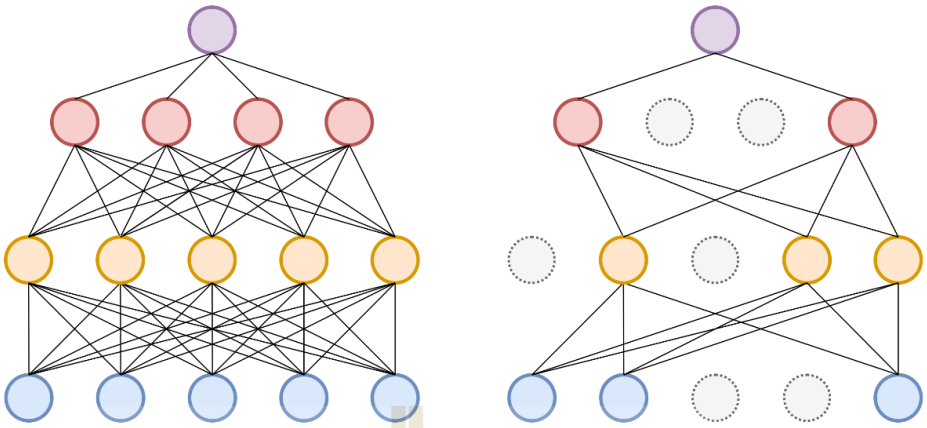


Figure 2.14 Dropout neural network model. (a) A standard neural network. (b) An example of a thinned net produced by applying dropout

2.7.3 Batch Normalization

In 2015, Batch Normalization (BN) was proposed by Ioffe and Szegedy (Ioffe & Szegedy, 2015). Batch Normalization is used to make training of neural networks faster, more stable and to mitigate the problem of internal covariate shift.

It is called "batch" normalization because, during training, it uses the mean and standard deviation (or variance) of the values in the current batch to level the values coming in from each layer. During training time, a batch normalization layer does the following:

$$\text{Batch mean}(\mu) = \frac{1}{m}\sum_{i=1}^{m} O_i \tag{2.14}$$

$$\text{Batch variance}(\sigma) = \sqrt{\delta + \frac{1}{m}\sum_{i=1}^{m}\left(O_i - \sigma_i\right)^2} \tag{2.15}$$

Where 0 is the previous layer, m is the number of samples in the given batch. σ is the batch standard deviation. By δ is a small number to make sure $\sigma > 0$ i.e., to make sure 0 from becoming undefined when you divide it by zero.

BN speeds up learning by letting people learn at much higher rates and by making initial learning rates less important. BN also sets a feature along with other features in each batch, which means that each normalized feature is not just a deterministic value. This effect lowers overfitting and sometimes gets rid of the need for other regularization methods (Srivastava, et al., 2014) like Dropout (Ioffe & Szegedy, 2015).

2.7.4 Softmax

In the final step of a neural network-based categorical classifier, particularly when dealing with more than two truth classes ($I > 2$), where *K* represents the number of classes, a softmax layer is commonly employed. The softmax layer plays a pivotal role in converting the model's raw output into a meaningful probabilistic interpretation (Fayek, Lech, & Cavedon, 2017).

$$\text{Softmax}\left(Z_i\right) = \frac{e^{z_i}}{\sum_{i=1}^{m} e^{z_i}} \qquad for\ i = 1, ..., I \tag{2.16}$$

Where K is the number of classes in the output and Z_i is the activation output for class i (Bishop, 2006). So, the SoftMax function takes the logits as input and performs a transformation that squashes the values between 0 and 1. Moreover, it also makes sure that the sum of all class probabilities equals 1.

2.8 Machine Learning

Machine learning is a branch of artificial intelligence that focuses on developing algorithms and models to learn from data and make predictions or decisions. The algorithms used for training are typically categorized as follows:

Supervised Learning: Supervised learning is a type of learning where data is provided with labelled examples and results. In this learning, the computer is trained to learn from input data and corresponding desired results provided by a teacher or supervisor. Next, the computer will link the data and generate a prediction model. In this type of learning, the algorithm learns from the labelled examples to generalize

patterns and relationships between input data and their corresponding outputs. Create a purposeful model to accurately predict results for new input data based on its learned knowledge. This learning is commonly used in tasks such as classification, where the model learns to assign input data into predefined categories, and regression, where the model learns to predict numerical values based on input features. Supervised learning is a widely applied approach in machine learning due to its ability to make accurate predictions with labelled training data (Nasteski,2017).

Unsupervised Learning: Unsupervised learning is where the computer receives input data without corresponding desired outputs or labeled examples. It aims to mimic the functioning of the human brain more closely. The learning process uses statistical principles to analyze and group the data into different levels. This learning doesn't rely on examples with labels but looks for information or patterns in the data itself. Unsupervised learning is especially helpful when there is no labeled data. The method is used in various domains, including exploratory data analysis, recommendation systems, and anomaly detection (Celebi & Aydin, 2016).

Reinforcement Learning: Reinforcement learning is a learning approach that involves trial and error to learn and determine the most effective course of action. It is often used when an agent learns how to interact with an environment to maximize a reward signal. Examples include learning to play games or optimizing product recommendations, predicting customer behavior, etc. In reinforcement learning, the agent learns through interactions with the environment. The model will receive feedback through rewards or penalties based on its actions after the model learns and improve to seek maximum overall reward to achieve an optimal solution. However, this learning allows the agent to learn and improve without humans, even when the data is complex and non-systematic data (Vidyasagar,2023).

While supervised learning, as discussed in the previous section, encompasses a range of techniques and models, one of the most effective and widely used in this category is the SVM (Kadiri & Alku, 2019). This method stands out for its unique

approach in solving classification problems, leveraging the principles of supervised learning to achieve high accuracy and efficiency.

2.9 Support Vector Machine (SVM)

The Support Vector Machine (SVM) technique is popularly applied to classification problems (Wang,2005). It is built upon the foundations of linear models and is particularly effective for grouping data. It finds the best-separating hyperplane with a wide margin that touches the closest data points in the feature space. As a result, the hyperplane with the most significant margin is regarded as the best-separating boundary, and the data points that contact this margin are known as support vectors, as depicted in Figure 2.15.

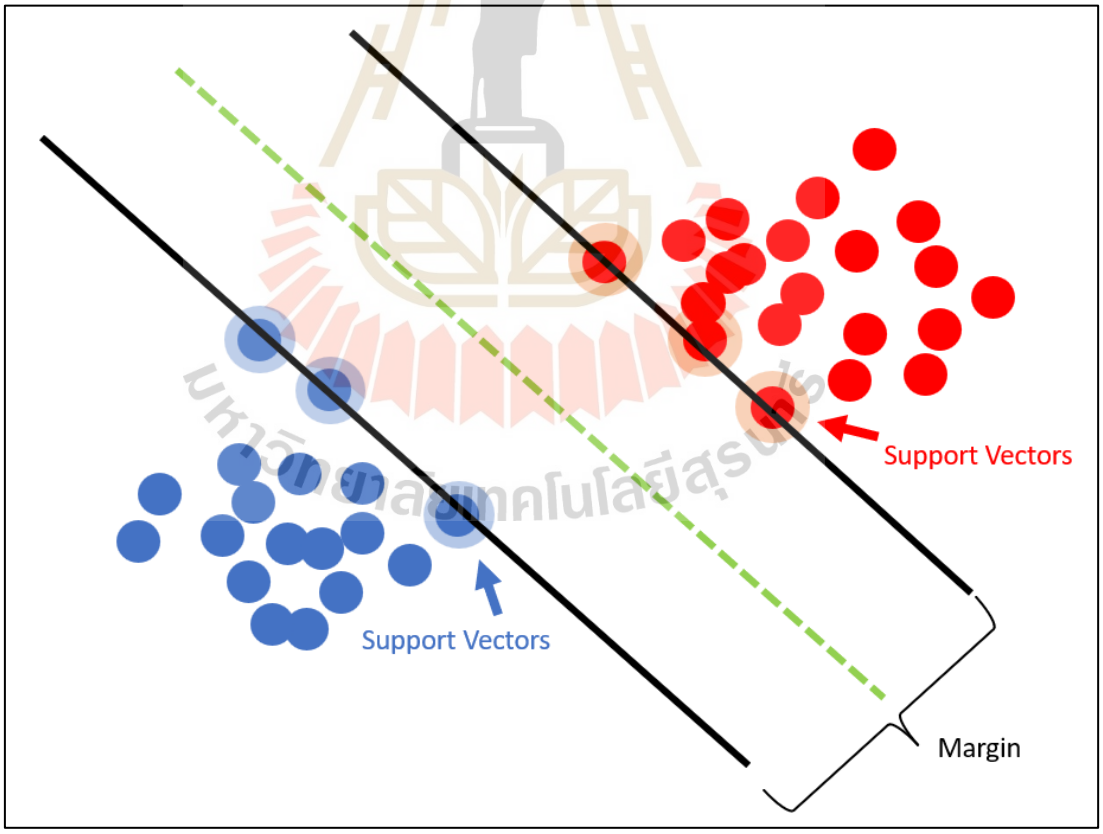


Figure 2.15 Support Vector Machine

Building on the foundational concept, the SVM is renowned for its geometric interpretation and adaptability. It is a potent supervised learning tool suitable for classification and regression tasks (Shen, Changjun, & Chen, 2011). Central to its adaptability are its hyperparameters, particularly C and γ.

C Parameter: The C parameter, colloquially known as the regularization parameter, dictates the balance between the enlargement of the margin and the reduction in classification error. It plays a pivotal role in harmonizing the objectives of minimizing training error while striving for a lower test error, thereby impacting the SVM's generalization potential. A lean towards a more minor C would inherently maximize the margin, albeit at the risk of misclassifying certain data instances. Conversely, escalating the value of C exhibits a determination to flawlessly classify training samples, possibly at the cost of opting for a hyperplane with a reduced margin. To encapsulate, C governs the penalty meted out for misclassification. An augmented C enforces a stringent punishment, refining the decision boundary to envelop the data points closely.

Gamma (γ) Parameter: Kernel Coefficient for “rbf”, “poly”, and “sigmoid”. It is intrinsically linked with the Radial Basis Function (RBF) kernel and finds relevance in the “poly” and “sigmoid” kernels. It is the torchbearer for defining the contour of the decision boundary. A subdued γ value magnifies the influence of individual training examples, paving the way for a malleable decision boundary. In stark contrast, an elevated γ confines the training example's impact to its immediate surroundings, giving rise to a more undulating decision boundary. Viewed through the prism of the RBF kernel, γ inversely correlates with the sphere of influence wielded by the training samples. The “scale” alternative for γ calibrates it based on the equation 2.17.

$$scale = \frac{1}{N_{features} \times \mathrm{var}(X)} \tag{2.17}$$

Wherein $N_{features}$ represents the feature count and $\mathrm{var}(X)$ denotes the dataset's variance. $\mathrm{var}(X)$ based on the equation 2.18

$$\mathrm{var}(X) = \frac{\sum_{i=1}^{m}(x_i - \mu)^2}{m} \tag{2.18}$$

Wherein x_i is the i^{th} observation in the dataset, μ is the mean (average) of all observations in the dataset and m represents the total number of observations or data points in the dataset X

The judicious selection of C and γ holds the key to the SVM model's performance. In real-world applications, practitioners often resort to methods such as grid search or random search, dovetailed with cross-validation, to pinpoint the optimal parameter values. Building on this understanding, when configuring our SVM with the RBF kernel, it becomes evident that two hyperparameters are of primary importance: C and γ.

The C parameter plays a crucial role in defining the balance between margin width and misclassification in SVM. Specifically, a small value of C prompts the SVM to prioritize a wide margin, possibly at the cost of some misclassifications in the training data. This might result in a more generalized model with a smoother decision boundary. On the flip side, a larger C value drives the SVM to reduce training misclassifications, even if it means a narrower margin. This can produce a complex decision boundary that fits the training data closely, but with a potential risk of overfitting.

The γ parameter in SVM. When set to low values, the influence of individual training samples becomes more widespread, leading to a smoother decision boundary. Such a configuration can often result in a model that is more generalized, capturing broader patterns in the data. Conversely, high gamma values create a contrasting effect. A high gamma value means that the influence of the training samples is more localized, potentially producing a more intricate, wavy decision boundary. This configuration can fit the training data very closely, but it comes with a heightened risk of overfitting. Delving deeper into its mathematical significance, in the context of the

RBF kernel, gamma essentially determines the distance over which two samples are considered “similar”. A high gamma, for instance, results in a narrower bell-shaped curve for the RBF kernel, implying that samples need to be near to be deemed similar.

2.10 Deep Hybrid Learning for pathological voice detection.

In deep hybrid learning, various machine learning methodologies are integrated, with the nuanced feature extraction capabilities of deep learning being combined with the classification prowess of traditional algorithms like SVM (Khairandish et al., 2021). This combination allows for a more nuanced and comprehensive understanding of data, particularly when dealing with complex and layered datasets. The significance of deep hybrid learning lies in its ability to capitalize on the strengths of its constituent parts while mitigating their individual weaknesses. For example, deep learning models excel at parsing and interpreting raw, unstructured data, but they often require substantial data and computational power (Thuwajit et al., 2022). On the other hand, traditional algorithms, while being more efficient with smaller datasets, might struggle with the high-dimensional data that deep learning thrives on. Hybrid learning models aim to bridge this gap, offering a balanced and efficient approach to solving machine learning problems.

One of the most notable applications of deep hybrid learning is in the field of medical diagnostics, specifically in the detection of pathological voice disorders. Voice disorders can be subtle and vary greatly among individuals, making them challenging to diagnose accurately. Deep hybrid learning models can process and analyze the nuanced variations in voice data, distinguishing between healthy and pathological conditions with a high degree of accuracy. This intricate process of differentiation and analysis is made possible by the unique architecture of deep hybrid learning systems, which employ a combination of advanced neural networks and classification algorithms.

2.10.1 Architecture of Deep Hybrid Learning systems

1) Feature extraction using CNN: The first stage involves a CNN architecture that processes raw voice data, represented as time-series frames. This CNN, often with multiple convolutional layers, is adept at automatically extracting a rich set of features from the raw input without the need for manual feature engineering.

2) Enhancement with SVM classifiers: Extracted features from the CNN are then fed into an SVM classifier. SVMs are known for their effectiveness in high-dimensional spaces and their ability to find the optimal boundary between classes with a maximum margin, which is crucial for medical diagnosis where the distinction between healthy and pathological samples is often subtle.

2.10.2 Detailed workflow of a Deep Hybrid Learning model for voice pathology

1) Input processing: Audio samples are first segmented into frames, which are then transformed into a suitable form, such as spectrograms, for CNN processing.

2) CNN feature learning: The CNN layers apply various convolution and pooling operations to the input, progressively abstracting and enhancing salient features. Each convolutional operation is defined by kernels of specific sizes, followed by pooling operations that reduce dimensionality and focus on the most relevant features.

3) Integration with SVM: The high-level features extracted by the CNN undergo dimensionality reduction, typically through a global average pooling operation. These condensed features, which retain the essential information, are then presented to the SVM classifier.

4) Classification and decision making: The SVM classifier processes the features to classify the voice as healthy or pathological. The decision is based on the learned hyperplane that best separates the feature space into distinct classes.

2.10.3 Implementation and training

1) Model training: The training process involves an initial phase where the CNN layers are trained using a labeled dataset. This dataset consists of a variety of voice recordings, including those from individuals with various voice disorders.

2) Integration and fine-tuning: Post-CNN training, the extracted features are used to train the SVM classifier. This phase may include fine-tuning the CNN in conjunction with SVM training to better align the feature extraction with the classification goals.

3) Testing and validation: The fully trained hybrid model is then validated and tested using separate datasets to ensure its diagnostic accuracy and generalizability to unseen data.

2.11 Particle Swarm Optimization (PSO) as Feature Selection

Feature reduction is an essential preprocessing technique in the field of classification. The primary objective of feature reduction is to decrease the dimensionality of the dataset while maintaining or improving classification performance compared to using the whole set of features. In general, feature selection aims to identify a minimal subset of features that is sufficient for solving classification problems. This is achieved by eliminating redundant and repetitive features from the original dataset. By applying feature selection as a data preprocessing step, it is anticipated that the less complex dataset will aid in training a classifier that is simpler, more efficient, and more accurate than if all features were used. As seen from the above definitions, feature selection has two main objectives: optimizing classification performance and minimizing the number of selected features.

Among the various techniques employed for feature selection, Particle Swarm Optimization (PSO) (Mallenahalli & Sarma, 2018) has emerged as a powerful and efficient method. PSO is a computational technique that draws inspiration from the collective behaviors observed in bird flocking (Kennedy,1995). Thus, this thesis also

uses PSO for feature selection. Compared to Genetic Algorithms (GA), another popular choice for feature selection, PSO offers several unique advantages and some disadvantages, as detailed in Engelbrecht (2007) and Eberhart and Shi (2001):

Advantages:

1) Speed: PSO often finds a solution faster than GA because it requires fewer parameter adjustments (Eberhart and Kennedy, 1995).
2) Simplicity: This method is easier to use as it does not involve operations like crossover and mutation, which are necessary in GA (Engelbrecht, 2007).
3) Flexibility: PSO is easy to implement for various optimization problems without significantly modifying its operational framework (Poli, 2008).

Disadvantages:

1) Local Minima: PSO can sometimes get trapped in local minima, especially in highly complex search spaces (Kennedy, 1997).
2) Dependency on Parameters: While fewer, the parameters such as the number of particles and inertia weight are critical and can significantly affect the performance (Shi and Eberhart, 1998).

PSO is an optimization technique that involves the utilization of a population of particles, collectively referred to as a swarm, to solve a given issue. Every particle navigates across the search space to find the optimal solution by updating its position and velocity. Specifically, the current position of a particle is represented by the vector $x_i = \left(x_{i1}, x_{i2}, ..., x_{iD}\right)$, where D is the search space's dimension. The locations are updated by the utilization of an additional vector, referred to as velocity $v_i = \left(v_{i1}, v_{i2}, ..., v_{iD}\right)$, which is subject to a predetermined maximum velocity, $v_{\max}$ and $v_{id} \in \left[-v_{\max}, v_{\max}\right]$. Throughout the process of searching, every individual particle retains a record of its optimal location, referred to as " $pbest$," as well as the optimal position of its neighboring particles, referred to as " $nbest$ ". In the scenario where each particle exchanges information with all other particles, it can be shown that all

particles possess an identical $nbest$, commonly called $gbest$. The following equations update the position and velocity of each particle:

$$v_{id}^{t+1} = w \times v_{id}^{t} + c_1 \times r_{i1} \times \left(pbest_{id} - x_{id}^{t} \right) + c_2 \times r_{i2} \times \left(gbest_{id} - x_{id}^{t} \right) \quad (2.19)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (2.20)$$

where t represents the t^{th} iteration in the search process, d is the d^{th} dimension in the search space, i is the particle index, w is the inertia weight, c_1 and c_2 are acceleration constants, r_{i1} and r_{i2} are uniformly distributed random values in [0,1], $pbest_{id}$ and $gbest_{id}$ represent the position entry of $pbest$ and $gbest$ in the d^{th} dimension, respectively. The general PSO procedure is depicted in Figure 2.16.

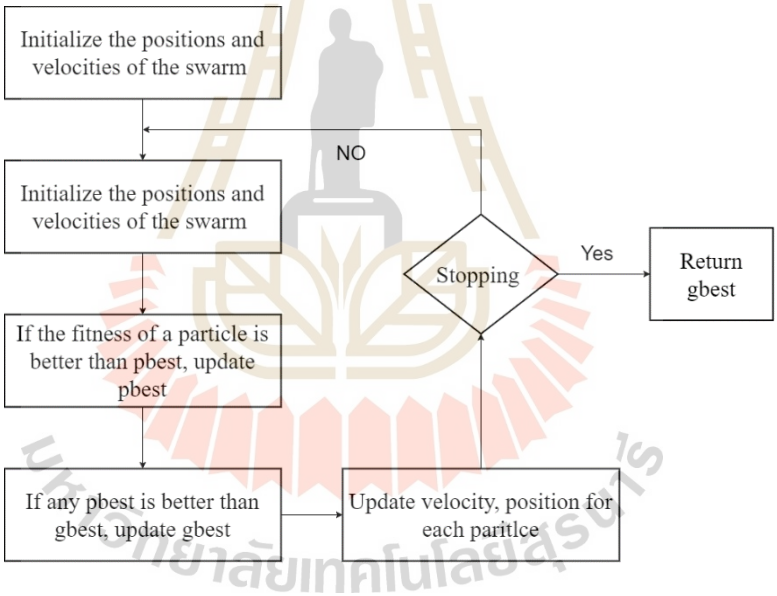


Figure 2.16 Flowchart of PSO

2.12 Confusion Matrix

When conducting experiments with various machine learning or deep learning models, it is crucial to have the ability to compare the effectiveness of these models. It will use three standard evaluation criteria suggested in Thuwajit et al. (2022) to determine how well our proposed methods work. The Confusion Matrix stands out as

a notable tool among the different methods employed to evaluate model performance (Luque, Carrasco, Martín, & De Las Heras, 2019). It is a prevalent and powerful tool that aids in evaluating accuracy and loss, providing valuable information on how well the model is making predictions. This matrix can give us deeper insights into the model's performance, making it a widely used technique. As seen in Figure 2.17, a confusion matrix is a 2x2 matrix. The model's performance and behavior become evident through the organized confusion matrix, which offers a clear understanding. Rows represent the predicted classes (the model output), and Columns represent the actual classes (reference data).

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Figure 2.17 Confusion Matrix for the binary classification

However, the most commonly used ones are accuracy (ACC), sensitivity (true positive rate, TPR) and specificity (true negative rate - TNR). They are calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.21}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{2.22}$$

$$Specificity = \frac{TN}{TN + FP} \tag{2.23}$$

When TP (True Positive) and TN (True Negative) are that, the fully trained network correctly forecasts the pathological and healthy voice classes, respectively. On the other hand, FP (False Positive) and FN (False Negative) are that the network wrongly labels a healthy voice as pathological and a pathological voice as healthy.

2.13 The t-Distributed Stochastic Neighbor Embedding (t-SNE)

The t-Distributed Stochastic Neighbor Embedding (t-SNE) method gives each data point a place in a two- or three-dimensional area so that high-dimensional data can be seen. This method, which builds on Stochastic Neighbor Embedding (Hinton & Roweis,2002), was made even better by Maaten and Hinton (2008), who made it t-distributed. t-SNE successfully lowers the dimensions of high-dimensional data so that it can be visualized. It does this by ensuring that similar data points are placed close together in the low-dimensional space and points not similar are set farther apart.

There are two main parts to how the t-SNE method works. First, it creates a probability distribution among the high-dimensional data points, giving points that are similar a higher chance of being true and points that are not similar a lower chance. Then, t-SNE makes a similar probability distribution in the low-dimensional space and tries to keep the Kullback-Leibler divergence between these two distributions as small as possible. The t-SNE has been used in many areas, such as genomics, computer security (Gashi, Stankovic, Leita, & Thonnard, 2009), natural language processing, music analysis (Hamel & Eck, 2010), cancer research (Jamieson et al., 2010), and biomedical signal processing (Birjandtalab, Pouyan, & Nourani, 2016).

For example, t-SNE can turn the Modified National Institute of Standards and Technology (MNIST) database of handwritten numbers into a two-dimensional map where each number is shown as a point. This image shows groups of similar numbers, showing how t-SNE effectively groups data points based on their similarity in a high-dimensional space. The t-SNE embeddings of the MNIST dataset are shown in Figure 2.18.

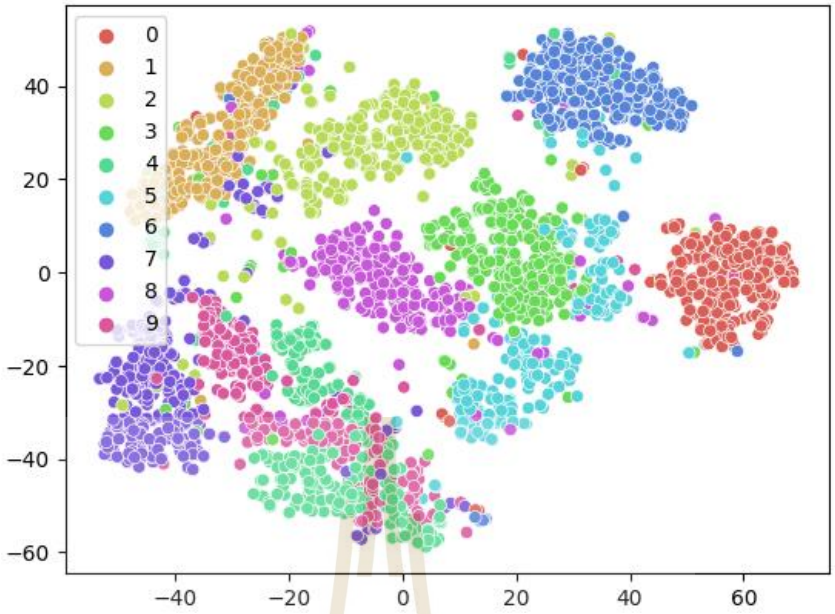


Figure 2.18 The t-SNE embeddings of MNIST dataset

2.14 Related work

This subsection gives an overview of state-of-the-art work on the datasets TORGO. Next, literature relevant to constructing the proposed architecture is presented, focusing on different types of artificial intelligence and the relationship between pooling and kernel size. The concept of local feature learning blocks and multi-scale convolution layers is also explored.

(Mesallam et al., 2017) This research presents a fully learnable audio front end that combines time-domain filter banks and Per Channel Energy Normalization (PCEN). This model is a pioneering approach to acquiring knowledge of extracting, compressing, and normalizing features from unprocessed waveforms in conjunction with a classifier. The model was applied to dysarthria detection and showed improved performance compared with fixed features. Learning the filters, normalization, and compression jointly with the architecture resulted in a 10% absolute accuracy improvement over fixed features. Learning only the time-domain filter banks or the PCEN parameters individually led to better results than fixed features. However, learning both jointly still

provided similar or better performance, demonstrating the potential of fully learnable audio front-ends.

(Narendra, Schuller, & Alku, 2021) This research investigates detecting Parkinson's Disease (PD) from speech using voice source information and two classifier architectures: the traditional pipeline approach and the end-to-end approach. In the traditional pipeline approach, SVM classifiers were developed using baseline acoustic features and glottal features extracted from speech utterances. The highest classification accuracy (67.93%) was achieved by combining baseline and Quasi-Closed Phase (QCP) based glottal features. Deep learning models were trained using raw speech waveforms and voice source waveforms in the end-to-end approach. The system trained using QCP-based glottal flow signals achieved the highest accuracy (68.56%). The study found that extracting voice source information was most effective in the traditional pipeline and end-to-end approaches. When voice source information was merged with baseline features, the accuracy of the SVM-based detection system trained with baseline features improved from 65 to 67.

(Sabir et al., 2017) This research introduces a refined algorithm tailored to detect and gauge the intensity of voice disorders in students by harnessing acoustical metrics with neural networks. With an impressive accuracy of 97.9%, a sensitivity of 1.6%, and a specificity of 95.1%, the algorithm differentiates between regular and abnormal voice patterns. It is a valuable tool for medical professionals and educators, facilitating continuous tracking of voice disorder progression in students, anchored in the acoustic characteristics of their speech. The algorithm can be applied in preventive medicine for early detection of voice pathologies. Furthermore, by juxtaposing observed parameters with benchmark values, the algorithm can reduce the severity gradient of voice ailments. The validation dataset encompassing healthy and impaired voice samples was sourced from a renowned German voice disorder repository.

(Vaiciukynas et al., 2014) This research introduces a new glottal inverse filtering technique called Quasi-Closed Phase (QCP) analysis, which performs a closed phase

type analysis over a time frame of multiple fundamental periods using Weighted Linear Prediction (WLP). It attenuates the contribution of the (quasi) open phase, resulting in an estimate of the vocal tract transfer function less influenced by the excitation. The method outperforms other Glottal Inverse Filtering (GIF) methods in objective measures obtained through inverse filtering synthetic vowel databases and subjective listening tests. However, the proposed methods have two constraints. Firstly, it requires accurate information about the Glottal Closure Instants (GCIs) to form the appropriate weight function, which may degrade its performance in real-world situations where accurate GCI data is unavailable. Secondly, the proposed method does not guarantee filter stability, which adds computational cost in GIF applications where all-pole synthesis is needed.

(El Emary, Fezari, & Amara, 2014) This research focuses on developing a voice pathologies detection system using acoustic voice analysis methods based on adaptive features, such as MFCCs with different Jitter and Shimmer. The research aims to identify different sound patterns of diseases, improve the capacity of voice features, and classify pathological voices using known techniques. The results show that a good classification rate is achieved with 39 coefficients, including Jitter and Shimmer, indicating that the difference between normal and abnormal becomes noticeable with the second derivative of MFCCs and energy. The number of Gaussians in the Gaussian Mixture Model (GMM) used as a classifier also affects the system's accuracy. The paper suggests the need for multivariate analysis of parameters and the importance of finding and sorting features that provide more information.

(Alhussein & Muhammad, 2018) This research investigates a voice pathology detection system using deep learning on a mobile healthcare framework. Voice samples are recorded via intelligent mobile devices, subsequently undergoing processing, and then channeled into a CNN. This research employs the transfer learning approach, harnessing the strengths of established CNN architectures, notably the VGG-16 and CaffeNet. Experiments utilize the Saarbrucken voice disorder dataset.

Experimental results show that the voice pathology detection accuracy reaches up to 97.5% using the transfer learning of CNN models.

(Harar et al., 2017) This research presents a preliminary investigation of voice pathology detection using deep neural networks and achieved promising results. The experiment used voice recordings of sustained vowel /a/ produced at normal pitch from the German corpus Saarbrucken voice disorder dataset, which contains voice recordings and electroglottograph signals of more than 2,000 speakers. The trained model achieved an accuracy of 71.36% with 65.04% sensitivity and 77.67% specificity on the validation files and an accuracy of 68.08% with 66.75% sensitivity and 77.89% specificity on the testing files.

(Janbakhshi, Kodrasi, & Bourlard, 2021) This research proposes a novel automatic dysarthric speech detection approach based on pairwise distance matrices and CNN. This method demonstrates enhanced performance compared to previous CNN-based models, but it also functions as an effective and dependable tool for diagnosing and managing clinical dysarthria. The effectiveness of this technique is additionally bolstered by experimental results obtained from databases that include healthy individuals and those with dysarthria, including a range of languages and situations. The proposed integrated framework improves feature extraction, distance matrix calculations, and the CNN classifier, providing a comprehensive solution for dysarthric speech.

(Narendra & Alku, 2020) research explored the role of glottal source information in identifying pathological voices by contrasting the traditional pipeline approach to the end-to-end approach. Employing glottal characteristics alongside openSMILE features, the conventional pipeline yielded promising results in pinpointing pathological voices. On the other hand, the end-to-end approach, leveraging deep learning models trained on glottal flow waveforms, outperformed models using mere raw speech. Within the conventional pipeline, merging glottal and acoustic features

enhanced the classification outcomes. The findings emphasize the significance of glottal source attributes in distinguishing pathological voices from typical ones.

2.15 Summary

The content mentioned above in this chapter delves into the basic neural network's theory. It also explores the essential concepts of feature selection. It emphasizes the importance of shared information as a metric to measure the connection between random variables. At the same time, it emphasizes the courage to distinguish non-linear feature interactions. The main aim of feature selection is to identify features closely associated with class labels. This discourse covers feature selection metrics such as distance, correlation, and consistency measures to ensure that duplicate information is excluded. It improves the quality of feature collection by factoring in differences such as variable dependencies. And identifying the optimal feature subset. Additionally, this chapter discusses the application of probability distributions and prediction sequences in the area of output labels. Including empty It describes in detail the calculation of output probabilities for every time step and the summarized output sequence.

Finally, several research papers have been conducted to investigate detecting dysarthric speech using machine learning and deep learning techniques. Thus, this thesis introduces a sophisticated methodology for identifying dysarthric speech. The forthcoming chapter will provide a comprehensive explanation of the intricate self-optimization process.

CHAPTER III

METHODOLOGY

3.1 Introduction

Voice analysis, especially pathological voice detection, has witnessed significant advancements in deep learning methodologies. While traditional methods have provided substantial insights, the increasing complexity of voice data and the need for more accurate for more innovative approaches. Recent literature has underscored the successes of the Multi-Scale Convolution Neural Network (MSConvNet) in various classification tasks. Its ability to explore multi-scale convolution blocks and extract multi-dimensional representations from data sets it apart. However, its potential in pathological voice detection remained, until now, an untapped area of research.

The steps in this section explain how to use the RS-MSConvNet model, a complete system designed to use the MSConvNet for finding pathological voices. This model diverges from traditional methods by processing raw speech data, eliminating the need for feature extraction, which often acts as a bottleneck in voice analysis tasks. The foundational principles behind MSConvNet, the rationale for its adoption, and the specifics of how the RS-MSConvNet model was architected will be elucidated in this section. This includes exploring the multi-scale convolution blocks, integrating spatial-

temporal feature blocks, and the final classification layers. Each choice is backed by rigorous theoretical underpinnings, which this chapter aims to illuminate.

3.2 RS-MSConvNet design

This subsection proposes a model tailored explicitly for the challenges of pathological voice detection. Rooted in the principles of multi-scale convolution, the RS-MSConvNet aims to harness the power of convolutional networks to scan voice data across various scales. By doing so, it aspires to capture a rich array of features, thus providing a comprehensive representation of voice patterns that could hint at pathology. The significance of this approach lies in its potential to detect features at different resolutions, acknowledging the fact that vocal patterns manifest across diverse temporal scales. Furthermore, the end-to-end nature of the RS-MSConvNet ensures that raw voice data can be processed directly, eliminating the need for manual feature engineering. The proposed model also envisages integration with a SVM classifier, culminating in a hybrid solution that seeks to merge the strengths of both techniques. This portion introduces the architectural subtleties and design decisions of the RS-MSConvNet, as seen in Figures 3.1 and 3.2.

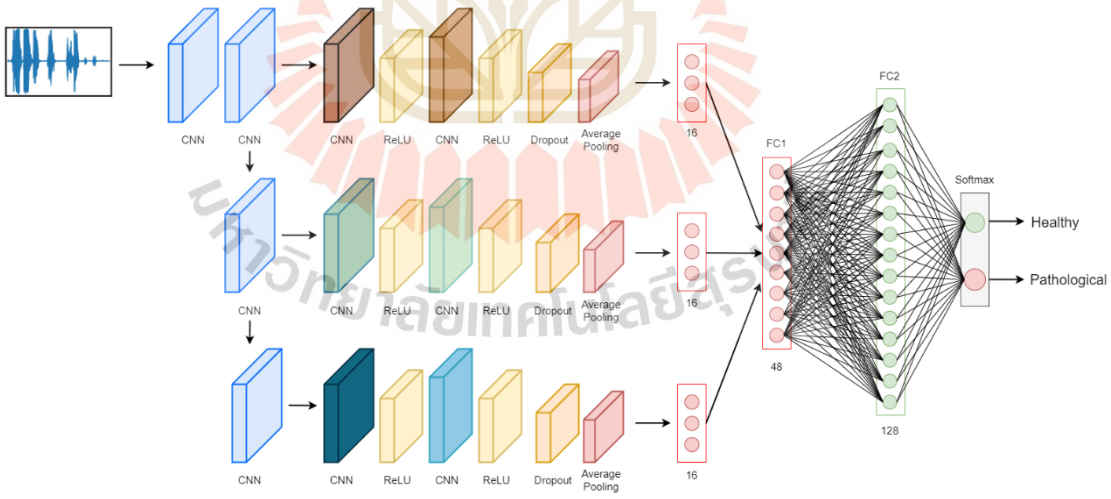


Figure 3.1 RS-MSConvNet

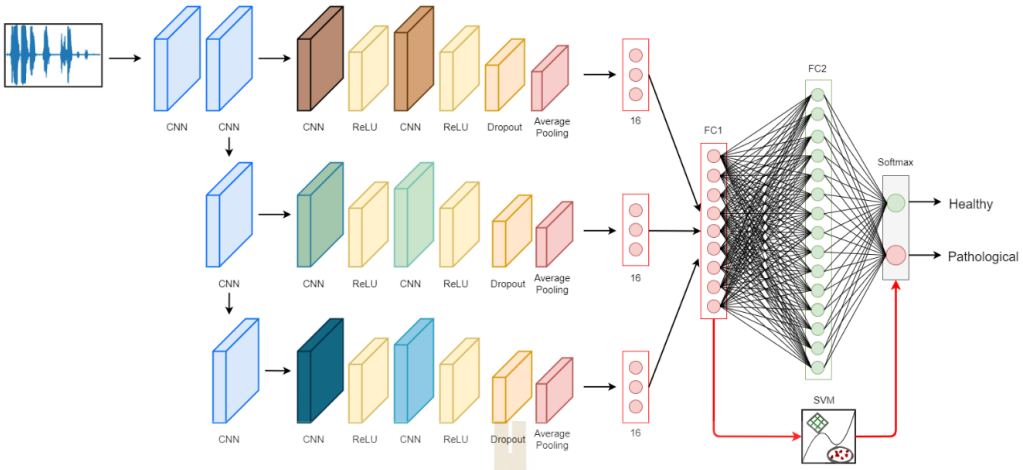


Figure 3.2 RS-MSConvNet-SVM

The RS-MSConvNet framework, which is suggested for the identification of pathological speech, comprises many components, including a pre-processing block, a multi-scale convolution block, a spatial-temporal feature extraction block, and a classifier block. These components are illustrated in the flowchart depicted in Figures 3.1 and 3.2. The setup of the RS-MSConvNet model is concisely outlined in Table 3.1.

Table 3.1 Configuration of RS-MSConvNet architecture, where (H,W) are the dimension of input representation and *k* denotes the order of layer in block b

Block	Layer	Kernel	Output	Activation	Parameters
b	Input		$(1,H,W)$		
	Conv2D	(2,2) Stride = 2	$\left(1,\frac{H}{2},\frac{W}{2}\right)$	Linear	5
	Conv2D	(2,2) Stride = 2	$\left(1,\frac{H}{4},\frac{W}{4}\right)$	Linear	5
	Conv2D	(2,2) Stride = 2	$\left(1,\frac{H}{8},\frac{W}{8}\right)$	Linear	5
	Conv2D	(2,2) Stride = 2	$\left(1,\frac{H}{16},\frac{W}{16}\right)$	Linear	5
c	Input		$\left(1,\frac{H}{2^k},\frac{W}{2^k}\right)$		

Table 3.1 Configuration of RS-MSConvNet architecture, where (H,W) are the dimension of input representation and *k* denotes the order of layer in block b (continued).

Block	Layer	Kernel	Output	Activation	Parameters
c	Conv2D	$32\times\left(\frac{H}{2^k},1\right)$	$\left(32,1,\frac{W}{2^k}\right)$		416
	Activation			Relu	
	Conv2D	$16\times(1,4)$	$\left(16,1,\frac{W}{2^k}-3\right)$		2064
	Activation			Relu	
	Activation			Dropout	
	Global average pooling		16		
d	Input		48		
	FC		128	Relu	
	FC		2	Linear	
	Classifier		1	Softmax	

3.2.1 Pre-Processing block

This section describes the systematic approach to generating suitable input data for training the RS-MSConvNet model. The process is initiated by implementing a pre-emphasis technique as the preliminary step to compensate for the high-frequency component of the speech signal's input. Next, the framing procedure is executed to structure the input data further. The raw speech signals are broken up into separate pieces called speech frames. Each frame has a frame length of 20 ms and a frameshift of 10 ms. A Hamming window is then used to improve the accuracy and consistency of these speech frames. This windowing process does two things: it strengthens the harmonics in each frame and smooths out the edges of each frame. The Hamming window has a transformative effect on the data, reducing possible distortions and flaws that could hurt the accuracy and general quality of the data. Finally, the raw information is transformed by arranging the segmented speech frames, which results

in a 2-dimensional (2D) arrangement. This organized dataset has been assembled to be the base for training the suggested RS-MSConvNet model. By setting things up this way, the model gets a full picture of the time order and the underlying differences in the data it uses. This big-picture view helps the model find and understand complex patterns and connections important to its learning process.

3.2.2 Multi-scale Convolution block

In this section, there has been a notable motivation derived from the works presented in references (Li et al., 2020; Ko et al., 2021; Janbakhshi et al., 2021). These works emphasize the importance of feature pyramid networks, especially when they are built upon the framework of a multi-scale convolution block. The Multi-scale convolution block is designed to extract semantic information across various scales. This feature is paramount because, in many scenarios, data carries different semantic values at different scales. The network can gather a more robust and comprehensive understanding of the data's semantic content by tapping into this multi-scaled information. Consequently, this enables the network to make predictions with greater precision. In essence, the system can achieve improved results across different application areas by harnessing the power of more robust semantic information obtained from scaled features.

The multi-scale convolution block is a cutting-edge approach tailored to handle 2D-input data. The primary purpose behind this implementation is to transform this 2D data into features that exist across a diverse range of scales. This kind of change is very important for giving the detection model the ability to see and learn about objects of all sizes, from very small to very large. Delving into the architectural specifics of our model, each multi-scale convolution block has been intricately crafted to extract and process input data. A notable aspect of this design is that every block operates at half the resolution scale of its preceding layer. This systematic downscaling ensures that the model can consistently capture finer details at each progressive layer, thus reinforcing its ability to comprehend data across a vast scale spectrum.

Moreover, an inherent intelligence has been built into these blocks. They are not just passive filters but are capable of active learning. Each block can autonomously determine the optimal weights during the training phase. This adaptive weight determination aids the block in sifting through the plethora of input data and pinpoints the most valuable level features. Doing so reduces the input signal by half, optimizing the process and ensuring that only the most crucial data characteristics are retained and emphasized in subsequent layers. This dual ability to discern the importance of features and reduce redundancy empowers our model to deliver exceptional performance in its tasks. This block defines the input data with a specific shape as (C, H, W). Here, each of these dimensions is characterized by:

C: This represents the number of channels in the input data.

H: This stands for the number of frames, signifying the sequential temporal chunks of the data.

W: This indicates the number of samples present within each frame.

Within this block, there's a series of 2D-convolution layers. Each of these layers performs a convolution operation on its input. The operation is standardized across all these layers, where the convolution is carried out using a kernel size of (2,2). Further specifications of this operation include a stride set at two and the absence of any padding. Such a configuration ensures that the convolution operation consistently reduces the input size. A direct implication of this design choice is witnessed in the relationship between consecutive convolution layers. Observing any given k convolution layer, the number of rows representing frames (H) and columns representing the samples in each frame (W) is precisely half of what they were in the $k-1$ convolution layer. The output size derived from the k layer is mathematically determined as $\left(1, \frac{H}{2}, \frac{W}{2}\right)$. An essential aspect to note is the utility of the outputs from these convolution layers. Specifically, the outputs generated from the second to the fourth layers are channeled into the subsequent block. This deliberate design allows for a deeper and more intricate extraction of spatial and temporal representation. The

rationale behind this choice is to leverage varying fields of view, enabling the architecture to discern features and patterns from multiple perspectives, thereby enriching its understanding of the input data.

3.2.3 Spatial-Temporal feature extraction block

In this block, the objective is to extract spatial-temporal features. These attributes arise from every individual scale of the output generated by the multi-scale convolution block. To facilitate this extraction, the design intricately employs two dedicated 2D-convolution layers. It is geared towards intercepting the last trio of scaled outputs that emanate from the block.

The first 2D-convolution layers utilize a trio of distinct kernel sizes of $\left(\frac{H}{4},1\right)$,$\left(\frac{H}{8},1\right)$, and $\left(\frac{H}{16},1\right)$. Each kernel is configured with a uniform stride of 2 and no padding. These kernels have been optimized to churn out 32 output channels. Their purpose is primarily to target and process outputs relayed from the second, third, and fourth layers. As for the second 2D-convolution layer in the sequence, it's uniquely characterized by a consistent kernel size of (1,4). Operating at a stride of 2 and no padding, it's designed to produce 16 channel outputs. Its main role is centered around gleaning and processing diverse outputs stemming from the block's introductory layer.

Finally, the global average pooling is applied to the output from the two regular 2D-convolution layers. This process results in a collection of 48 unique features, which can be broken down into 16 distinct features for every individual scale for easier understanding.

3.2.4 Fully Connected (FC) layer block

In the proposed model architecture, the Fully Connected (FC) Layer Block comes into play after applying two convolutional layers and a subsequent global average pooling operation. The spatial feature module's outputs, derived from various scales, are enhanced and channeled to the FC layers. A notable aspect of this architecture is the utilization of the Log softmax function, which serves as the activation function for these layers.

3.2.5 RS-MSConvNet: A detailed overview

In the evolution of CNN, the architecture and design of the model play an imperative role in its performance. This subsection delves into the architecture of the RS-MSConvNet, as delineated in Figure 3.3. A systematic will present each layer integral to the model's structure. The discussion will cover the distinct types and features of these layers, delve into their dimensional attributes, and enumerate both trainable and non-trainable parameters. Such an in-depth exploration is instrumental in comprehending the underlying mechanisms of the RS-MSConvNet and its potential implications in the broader realm of neural network research.

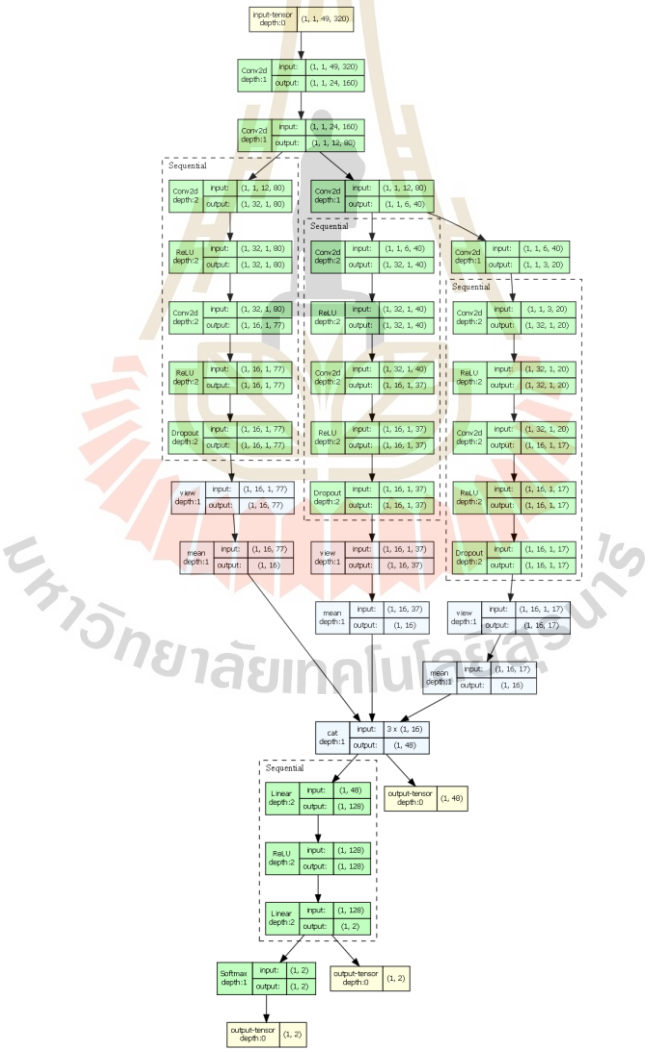


Figure 3.3 Comprehensive description of the RS-MSConvNet architecture

3.3 Resources

The models were built and implemented using Python 3.9.10, and the PyTorch v1.10.1 framework. The models were run using the following machine:

CPU: AMD Ryzen Threadripper 3970X (32 Cores, 64 Threads, 3.7 GHz)

GPU: NVIDIA RTX3090 (24 GB)

RAM: ZADAK Twist DDR4 (3200MHz, 4x32GB)

3.4 Experimental setup

3.4.1 Database

The TORGO database (Rudzicz, Namasivayam, & Wolff, 2012), utilized prominently in this study, stands out due to its challenges. One primary hurdle associated with TORGO is its comparatively limited dataset, especially when juxtaposed against other available databases. As demonstrated in reference (Narendra & Alku, 2020), models designed end-to-end on the TORGO database often find it challenging to surpass or even match the accuracy rates of those constructed on other databases.

The TORGO corpus is a remarkable outcome of a synergistic collaboration between two distinguished departments of the University of Toronto: Computer Science and Speech-Language Pathology. This partnership was further strengthened with the involvement of the Holland-Bloorview Kids Rehab Hospital, a renowned institution based in Toronto. The result of this alliance is a comprehensive public database that serves as a gold mine for researchers and professionals alike.

This rich database isn't just a mere collection of voice recordings; it's a carefully curated assortment representing various voice types and conditions. The participants whose voices have been captured in the database come from diverse backgrounds and health conditions. This includes three females diagnosed with dysarthria, labeled as F01, F03, and F04. In contrast, three other females, FC01, FC02, and FC03, have no such diagnosis and are considered to have typical voice patterns. The male participants add further depth to the database. Five of them, identified as M01 through M05, have been diagnosed with dysarthria. Meanwhile, five others, MC01 through MC04, exhibit standard voice characteristics.

Delving deeper into the technical aspects of the TORGO corpus reveals a consistency that underscores the meticulous planning and execution that went into its creation. Every voice utterance in the database has been sampled at a rate of 16 kHz, ensuring uniformity in the quality of recordings. The contributions from participants are equally noteworthy. Those without voice disorders, often called non-dysarthric participants, have made a significant contribution, averaging about 900 utterances each. On the other hand, despite their voice challenges, the dysarthric participants have been close behind, contributing an average of approximately 400 utterances each. This balance showcases the database's commitment to providing a holistic view of voice patterns, ensuring that typical and atypical voices are adequately represented, thereby enriching the depth and diversity of the corpus. The recorded utterances in this database encompass a diverse range:

1) **Non-words**: These non-words provide a baseline for evaluating the articulatory capabilities of dysarthric speakers, especially concerning plosive consonants and prosody. Examples include repetitions of phonetic patterns like /iy-p-ah/, /ah-p-iy/, and /p-ah-t-ah-k-ah/.
2) **Short words**: These are particularly beneficial for acoustic speech studies, eliminating the need for word boundary detection.
3) **Restricted sentences**: These are used in Automatic Speech Recognition (ASR) to harness lexical, syntactic, and semantic processing.
4) **Unrestricted sentences**: Participants were encouraged to spontaneously describe intriguing scenarios depicted on cards from the Webber Photo Cards: Story Starters collection. These sentences reflect the intricacies of genuine spoken language, including natural disfluencies and diverse syntactic structures.

Considering the unique structure of the TORGO database, it was observed that a significant portion of the recordings was dominated by silence. This necessitated the removal of these silent patches before proceeding with the training and testing of the classification model. In terms of data division for speaker-independent pathological voice detection, the database was categorized into three subsets:

1) Training Subset: Consisting of 3,125 healthy and 1,491 pathological utterances, totaling 3.5 hours.
2) Validation Subset: Comprising 944 healthy and 795 pathological utterances, cumulating to 2 hours.
3) Testing Subset: With 2,087 healthy and 861 pathological utterances, to 3 hours.

Table 3.2 comprehensively summarizes these subsets, highlighting their integral role in our experiments. Furthermore, the study's findings were meticulously contrasted against the experimental conditions detailed in (Narendra & Alku, 2020).

Table. 3.2 Details about three subsets of the TORGO database.

Training	Validation	Testing
MC03, MC04	MC02	MC01
FC02	FC01	FC03
M02, M05	M01, M03	M04
F01, F03	-	F04

3.4.2 Parameter tuning

In the pursuit of improving model performance and efficiency, selecting an appropriate learning rate plays a crucial role in the training process of models. Determining the optimal initial learning rate involves conducting initial experimentation with a range of values, typically from 0.1 to 0.00001. Through this experimentation, the goal is to identify the best learning rate. In this study, after thorough exploration and experimentation, it was concluded that our model's most effective initial learning rate is 0.0001. The learning rate of 0.0001 has emerged as the optimal choice, carefully evaluated based on its impact on both the attained accuracy and convergence time.

In the following step, preliminary experimentation was conducted to analyze the performance of several optimizers, such as Stochastic Gradient Descent (SGD) and Adam. The objective was to identify the optimal optimizer for our model. SGD was the optimal optimizer due to its high initial experimentation precision.

In the final step, During the model training process, two essential techniques, Dropout Rate and Early Stopping, were used as preventative steps to lower the risk of overfitting. Dropout helps make the network more stable by randomly turning off some neurons during each cycle. At the same time, early stopping ensures that training stops after a certain number of iterations when the validation loss stops improving. The model will improve its performance and prevent overfitting by taking these steps.

Table 3.3 Summarizes the model parameters for the RS-MSConvNet, RS-MSConvNet-SVM and RS-MSConvNet-SVM with PSO models.

Parameters	**RS-MSConvNet**	**RS-MSConvNet-SVM**	**RS-MSConvNet- SVM with PSO**
Optimizer	SGD	SGD	SGD
Batch Size	256	256	256
Learning Rate	0.0001	0.0001	0.0001
Decay	0.0001	0.0001	0.0001
Momentum	0.9	0.9	0.9
Dropout	0.5	0.5	0.5
Epoch	1000	1000	1000
SVM (C)	N/A	1	1
SVM (**gamma**)	N/A	0.1	0.1
PSO (W)	N/A	N/A	0.7
PSO (C_1)	N/A	N/A	1
PSO (C_2)	N/A	N/A	3

3.5 Summary

In this chapter, the RS-MSConvNet model represents a significant advancement in pathological voice detection, leveraging the strengths of deep learning and the innovative use of multi-scale convolution neural networks. By integrating a range of techniques, from pre-processing methods to spatial-temporal feature extraction, this

model offers a comprehensive and nuanced approach to analyzing voice patterns. Its design, which combines multi-scale convolution with a support vector machine classifier, showcases a unique and potentially highly effective method for detecting voice pathologies. Furthermore, a detailed experimental setup, including the TORGO database and sophisticated parameter-tuning strategies, underscores the model's robustness and applicability in real-world scenarios.

CHAPTER IV
RESULTS

4.1 Introduction

Voice detection and analysis is an evolving field, particularly notable for its advancements in pathological voice detection. The accuracy of this area is crucial, considering its significant clinical implications and potential to revolutionize voice pathology diagnosis. Recent research highlights the effectiveness of convolutional neural network, especially MSConvNet, in diverse classification tasks. Yet, the application of MSConvNet in identifying abnormal voice patterns remains an area ripe for exploration.

This section presents the results of the proposed RS-MSConvNet model, a novel end-to-end solution built on the MSConvNet model, designed specifically for using raw speech data to find voices that are not normal. The model's unique building blocks, such as multi-scale convolution blocks, spatial-temporal feature blocks, and a fully connected classification layer, are tested to see how well it's effective at finding pathological voices. This part of the thesis will look at the RS-MSConvNet's complex performance metrics, comparing them to current state-of-the-art models and pointing out the big steps forward in improving the detection of pathological voices.

4.2 Speech length optimization in RS-ConvNet

Finding the best segment length for fixed-length segments is a crucial part of speech recognition and processing that can significantly affect how well end-to-end networks work (Tirronen, 2022). It is important to stress this factor is essential since the segment length selected for processing can determine how well the network works. The main goal of this study is to thoroughly look into different fixed-length segments to find the segment lengths that produce the best results. Specifically, this thesis has chosen to delve into segment durations of 240 ms, 250 ms, 500 ms, 1 second, and 3 seconds. Fig 4.1 shows the results derived from the RS-ConvNet model's experimentation to aid in visualizing and understanding our findings.

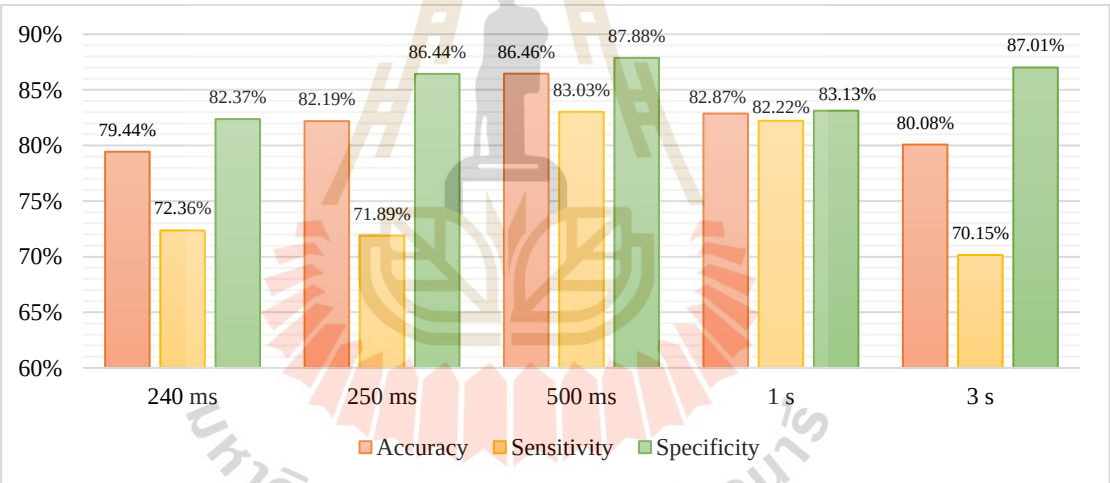


Figure 4.1 The RS-ConvNet classifier performance with different speech lengths

In Figure 4.1, It can be seen the results of tests that show how different fixed-length segments affect how well end-to-end networks work. When the tested features were compared, the one with a length of 500 ms showed the best performance, outperforming the others in terms of how well it worked. One problem was that segments with shorter distances, like 240 or 250 ms, had some issues. These shorter lengths make it harder for the network to process raw speech signals correctly. It might be harder for the network to learn from these short parts because they need to give more detailed information to pick out the subtleties of speech patterns that need to

be clarified. These limits could be bad, especially when trying to understand and tell the difference between the complicated changes in disordered speech patterns.

On the other end of the bandwidth, segments that last longer than 500 ms bring their problems. Significantly, these longer segments can be limiting, especially when considering how short some vowels are by nature, which has been talked about a lot (Tirronen, 2022). After looking at all of these points and, more importantly, in the context of our RS-MSConvNet model, it is clear that the 500 ms segment, which has a resolution of 49x320 pixels, is the best option.

4.3 Learning rate impact on RS-ConvNet

The learning rate is a hyperparameter that has a big effect on the complicated framework of deep learning (Shi,2020). The amount of the weight adjustment is significant because it controls how much the model weights are changed during training. An excellent way to think about optimizing this hyperparameter is as if it were walking carefully on a tightrope. Finding the right balance can help models come together faster and improve overall performance. On the other hand, making the wrong choices can cause training to go wrong and results not to be up to par. This study starts a planned investigation by looking at what happens when the learning rate for the RS-MSConvNet classifier changes. This will help to comprehend the subtleties of this delicate balance. By looking at a variety of learning rates, this thesis hopes to find information that will help it pick the best learning rate, which will make the model work better overall. The results are shown in Figure 4.2.

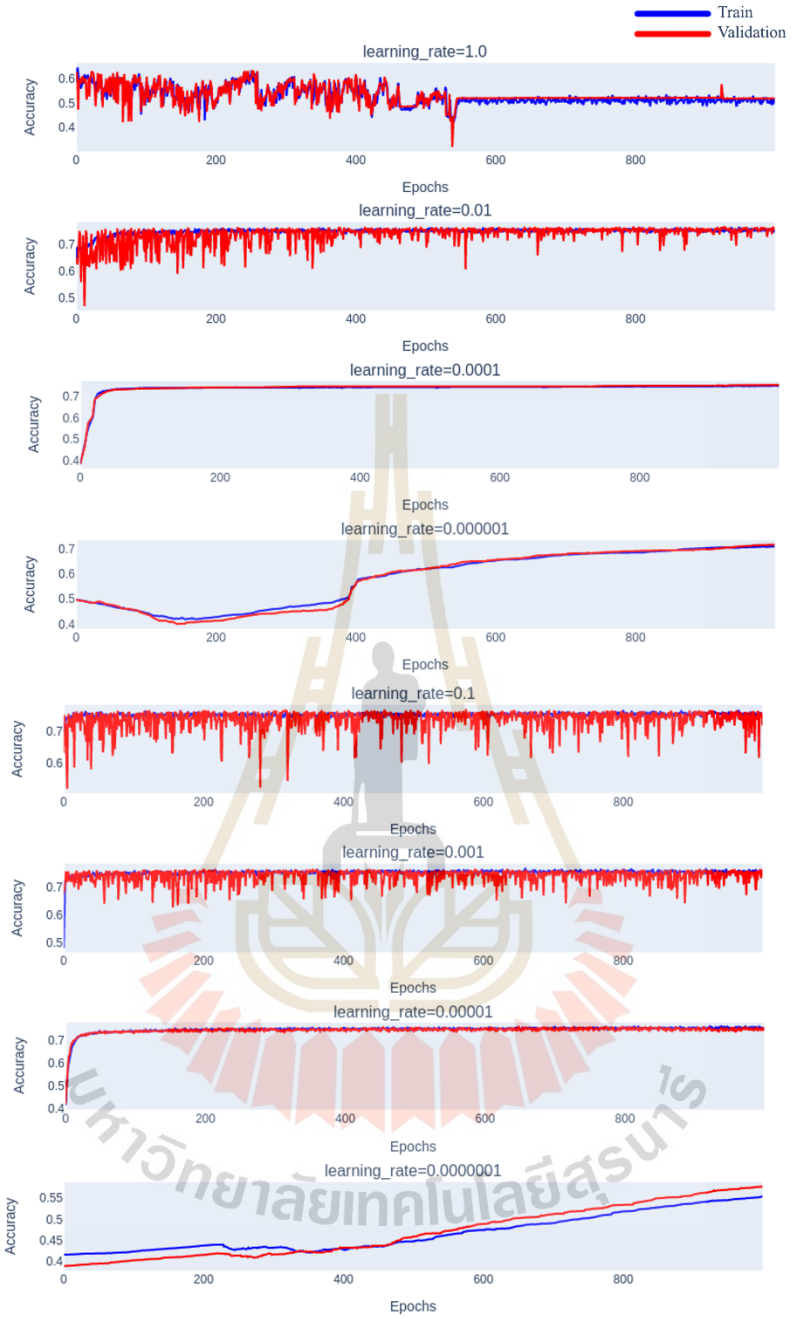


Figure 4.2 The RS-ConvNet classifier performance with different learning rates

As seen in Figure 4.2, it divides learning rates into three separate groups, each with its own set of values. To properly calibrate deep learning models, it's necessary to know the specifics of each layer, which represents a different level of learning speed. This tier-based approach makes it clear what each learning rate category means so

that one can make an informed decision that can improve the accuracy and efficiency of the model. The differences between these three groups, shown visually in Figure 4.2, are explained below.

Low learning rates: It was clear that the model was convergent slowly when very small learning rates like 0.0001, 0.00001, 0.000001, and 0.0000001 were examined. Over many epochs, the model's performance measures got better over time. Still, the model had to be trained for too long before it was as good as other configurations. One interesting thing about these very low learning rates is that they move carefully through the lost environment. This in-depth research could lead to better model generalization. However, this possible benefit is cancelled because it makes computing more difficult.

Moderate learning rates: Learning rates between 0.01 and.001 were the best fit for the model's training in the middle range. The convergence time was faster than the lower level, and the model had the highest level of accuracy among the rates that were looked at. This balance encourages effective learning while also building protections against the risks of skipping over the best points in the loss landscape.

High learning rates: The training experience became less stable as the learning rate went from 0.1 to 1.0. There were big changes in the model's loss graph, which showed that it often strayed from the best weight configurations. At the start of the training process, performance quickly improved, but this momentum was quickly cut short by the unpredictable environment of the training, leading to a negative conclusion.

The experiment and the subsequent analytical discussion show how important the learning rate is when deep learning models are being trained. The observations support the principle of moderation, which says that learning rates between 0.01 and 0.001 are good.

4.4 Batch size effects in RS-ConvNet

Within the complicated field of deep learning, picking the right batch size is a critical factor that affects many aspects of training the model. Although the learning rate is vital for model calibration, it is impossible to overstate how important the batch size is (De,2016). The convergence trajectory can be changed, model performance can be affected, and training efficiency can be controlled. Using the RS-MSConvNet classifier as an example, this section starts a detailed investigation into the complex relationships between batch size and model performance. It is known that both small and large batch sizes have advantages and disadvantages. The experiment includes a lot of different batch sizes and carefully studies how they impact the RS-MSConvNet classifier both by itself and as a whole. Extensive tests reveal a wealth of information about the complex relationship between batch size and model performance. This deep look has revealed clear patterns and trends through this meticulous investigation. The RS-MSConvNet classifier's performance with different batch sizes is better understood with these results. Figure 4.3 shows information on batch sizes.

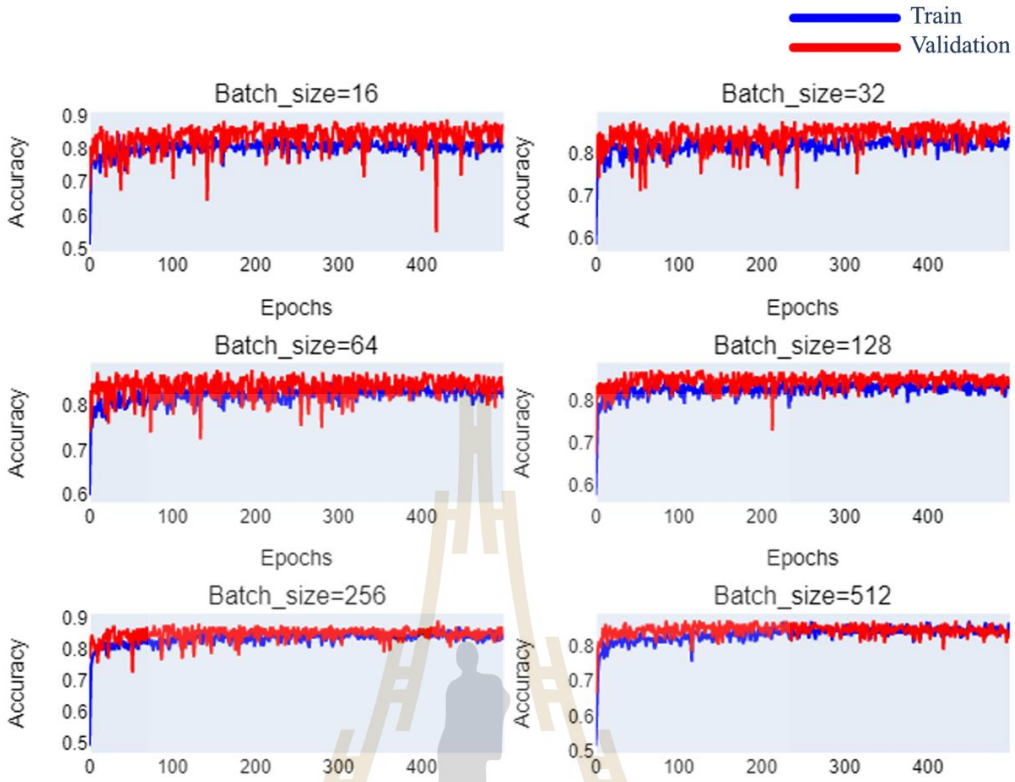


Figure 4.3 The RS-ConvNet classifier performance with different batch sizes

As depicted in Figure 4.3, the comprehensive analysis evaluated the model's performance across an array of batch sizes: 16, 32, 64, 128, 256, and 512. The empirical results underscore batch size's paramount significance in influencing model performance. The experimental findings show that batch size has the most significant effect on model performance. The data shows a clear link between batch size and the subtleties of how training works. Of all the batch sizes that were looked at, batch size 256 stepped out because its performance metrics were much better than its competitors. There are many good reasons to choose this particular batch size. It strikes a good balance between making sure that computations are quick and that the models are correct. This optimal size makes good use of computing power without weakening the reliability of model results. At the other end of the spectrum, smaller batch sizes have a catch, even though they offer stable and consistent convergence. Its

computational needs are higher, so they often have to be trained for longer periods, which may only be practical in some situations. It was the model's performance that got worse as batch sizes got bigger, especially when they went over 256. It's nice to have shorter computation times, but these bigger batch sizes have trouble applying to data they have yet to see.

4.5 Momentum dynamics in RS-ConvNet

There are many ways that momentum affects how models are trained and how well it does. Momentum is a key part of many optimizations' algorithms because it powers convergence and helps you find your way through the complicated parameter space (Shi,2021). Different optimization algorithms use momentum all the time, yet it still need to gain knowledge of how changing momentum values affect model performance. This is an area that needs more research. The main goal of this section is to make it clear what changing the momentum value means for the performance of a certain model. There are a lot of tests and evaluations done to find the best setting for momentum and to show how different values for momentum can change how well, accurately, and naturally a model works.

Finding the best model performance is like a complicated dance where it has to fine-tune many parameters. Momentum is one of the most important parts of this dance. The momentum value it chooses has a big effect on the learning algorithm. It determines how quickly it converges, how stable the training becomes, and whether it avoids problems at local minima. For the best model performance, it is important to find the momentum value that strikes the perfect balance between fast convergence and unwavering strength. Figure 4.4 provides insights into Momentum.

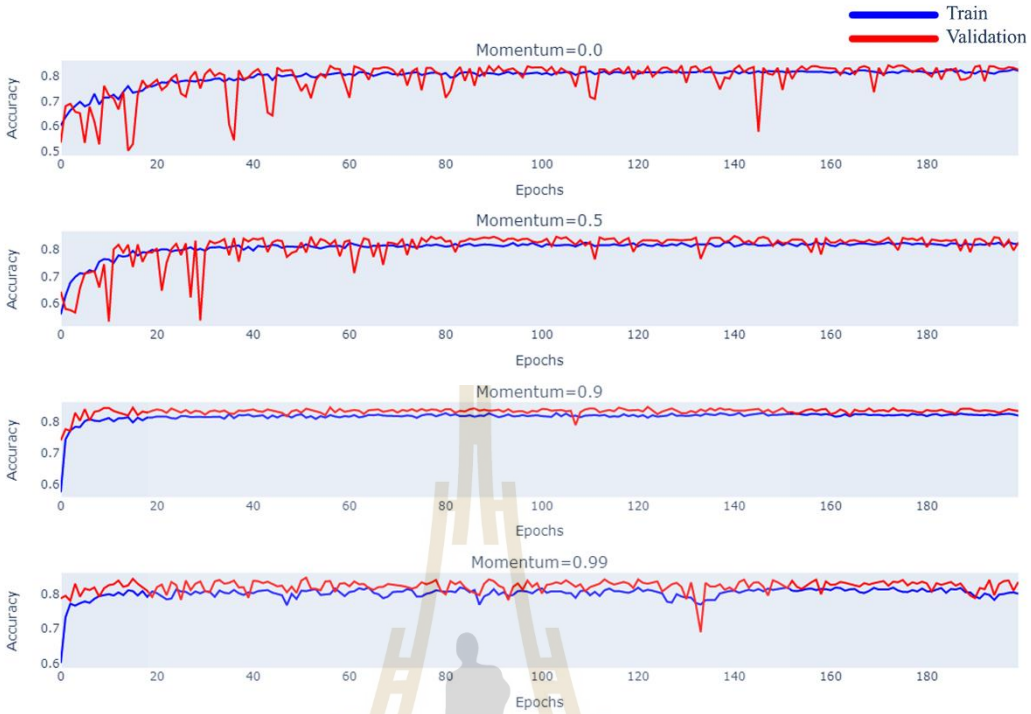


Figure 4.4 The RS-ConvNet classifier performance with different Momentums

Figure 4.4 shows that the values of momentum it looked at were all over the place: 0.0, 0.5, 0.9, and 0.99. The range of this bandwidth, from no momentum to very high momentum, was chosen to show the effects of momentum as a whole. The in-depth testing process made momentum's effect on the model's performance dynamics clear and noticeable. Out of all the values, a momentum of 0.9 came out because it had performance metrics miles ahead of the others. This shows that it is good at making the model's learning process run more smoothly and efficiently. As the group looked more closely at the data, a momentum value of 0.9 stood out as a sign of stability and effectiveness. It pushed for improvements in both the speed of convergence and the accuracy of the models. Lower momentum values, like 0.0 and 0.5, had a more leisurely convergence trajectory and training dynamics that were not as stable. Looking the other way, a momentum of 0.99 looked like it would lead to fast initial convergence. However, it showed signs of instability later in the training process, making it a potentially risky choice for the model being looked at.

4.6 Decay rate influence on RS-ConvNet

The decay rate is one of the most important factors for finding the best learning rate among all the others (You,2019). It is one of the most important parts of ensuring the training plan is stable and effective. Choosing the right decay rates can affect the rate at which deep learning models can converge, and it can generalize. It can improve model performance if it is adjusted correctly, making sure that the models are reliable and strong in a wide range of situations. Given how important decay rates are for training models, this section starts a thorough, organized look at how the selected model works with a wide range of decay rates. It is the goal of this in-depth study to learn more about how the different decay rates change the learning path, the speed at which the model converges, and its overall performance metrics. Figure 4.5 shows the decay rates of the impact.

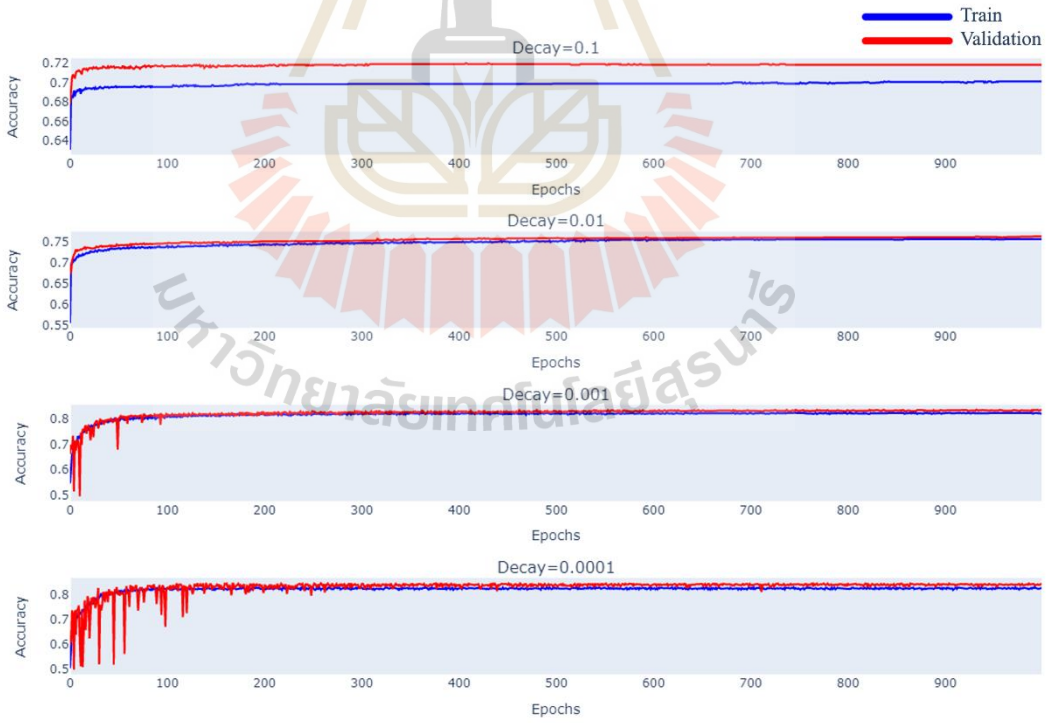


Figure 4.5 The RS-ConvNet classifier performance with different decay rates

In Figure 4.5, the effects on performance of different decay rates (0.1, 0.01, 0.001, and 0.0001). The research's careful analysis has revealed several important insights that are very helpful for finding the best decay parameter for the best model performance. Out of all the decay rates that were looked at, the rate of 0.01 stood out as the best for training the model. This discovery shows how important it is to improve the model's ability to learn quickly and well. Researchers found that a decay rate 0.01 was the best compromise between learning speed and stability. As the move towards the edges, a higher decay rate of 0.1 speeds up the learning process, but it did so at a price of model stability and its ability to generalize. This quick method resulted in less-than-ideal performance metrics, particularly as the model went through datasets it had never seen before. However, the less dangerous decay values (0.001 and 0.0001) slowed down the learning process while keeping the model stable, which is a good thing. The model could be better at working in real-life situations where things change quickly because it has a very slow learning rate.

4.7 FC layer effects in RS-MSConvNet

The FC layers have a significant effect on the way a neural network works for classification. The FC layer is the last part of the network. It is where all the extracted features are combined and processed to make the final result. This section goes into this vital detail and looks into how the number of FC layers affects the network's ability to identify pathological voices. It looked at how changing the number of FC layers in a neural network can change how well it can find and label voices that are not normal. These tests aimed to carefully check how well neural network models worked with various sets of FC layers. It tried combinations of one to five FC layers and looked at how each affected the network's ability to diagnose problems. It was able to see the subtleties of network behavior at different levels of complexity by using this method. The first model made had a neural network with only one FC layer. As the process went on, more and more layers were added. Each model underwent the same rigorous

testing process, ensuring our results were consistent and reliable. The results are shown in Figure 4.6.

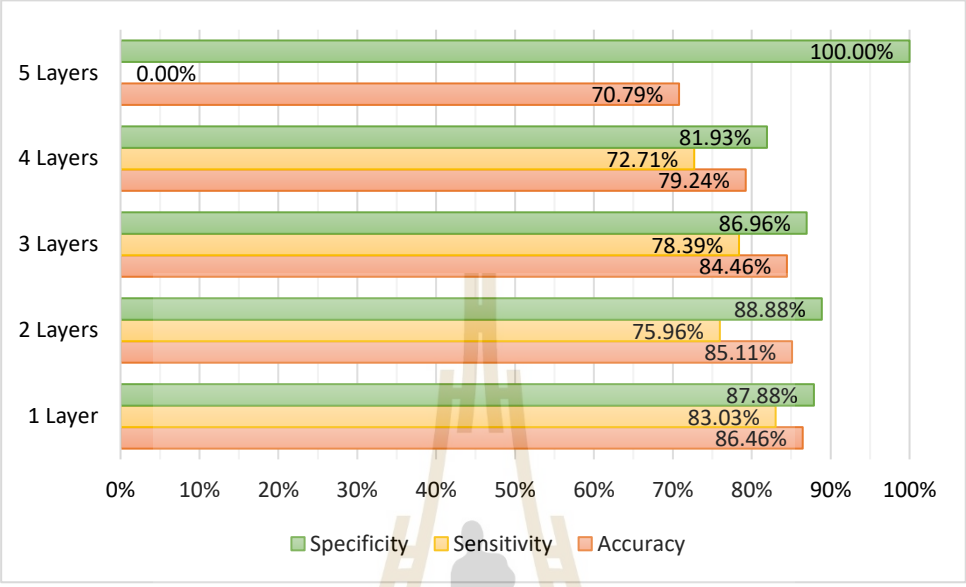


Figure 4.6 The RS-ConvNet classifier performance with different layers

In Figure 4.6, the model became less accurate when there were more than two FC layers. This exciting observation comes from sorting things into groups. The objective is to distinguish between two groups. For best performance, an FC layer configuration that is simpler and more streamlined, ideally only one layer, is recommended. Simpler architectures often do better than more complex ones when no training data exists, according to studies (Wang,2017; Phapatanaburi,2017)

4.8 Feature visualization in RS-MSConvNet

This section gives a detailed visual analysis of the features that can tell the difference between healthy and unhealthy voices using the best-configured RS-MSConvNet model. It includes the convolution layer outputs and the t-SNE method for a complete picture of voice signal properties.

4.8.1 Analyzing Convolution layers.

The RS-MSConvNet model was utilized to process both healthy and pathological voice signals. By examining the output representation from the second to the fourth convolution layers, it could visualize the differences in feature information. These layers offered varying resolutions 10×80 pixels, 5×40 pixels, and 2×20 pixels, respectively - each providing unique insights:

1) Second layer analysis (10×80 pixels): This layer began to show the basic structure of the voice signals, telling the difference between healthy and pathological voices.
2) Third layer analysis (5×40 pixels): More specific features started to show in this layer, making it easier to tell the difference between the two voices.
3) Fourth layer analysis (2×20 pixels): The most detailed layer highlights minor differences necessary for accurately detecting pathological voices.

Figure 4.7 shows the different images these layers create, highlighting the model's ability to distinguish between healthy and unhealthy voices with similar amplitude signatures. Using the matplotlib function (Hunter, 2007), the finer features of these representation images become even more transparent.

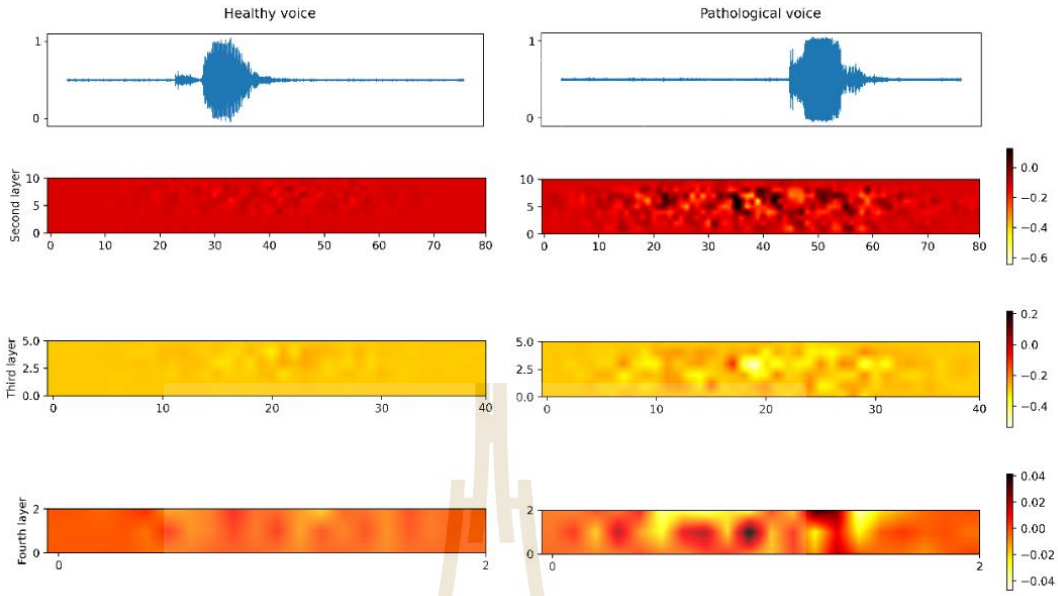


Figure 4.7 The outputs from the second to the fourth convolution layer are shown next to each other. Both come from patient #4, who is speaking “Train.”

In Figure 4.7, the model can tell the difference between healthy and unhealthy voices, even when the signals amplitude signatures are similar. The different representations seen across the convolution layers prove that there are unique features and that the RS-MSConvNet model's multi-scale convolution block works well.

4.8.2 t-SNE in Spatial-Temporal Analysis

The t-SNE (Van der Maaten,2008) technique was used to reduce the number of dimensions, focusing on the voice category distributions, so that the model's ability to tell the difference between categories could be better evaluated. The analysis used two hundred samples of unhealthy and two hundred samples of healthy voices to make it easier to see how the classes were distributed.

1) Distribution of raw speech signals (Figure 4.8a): The first used raw speech signals without feature extraction. It shows a lot of overlap in the data distributions, which made it hard to tell the difference between the voices.

2) Spatial-Temporal feature distribution (Figure 4.8b): The RS-MSConvNet model's spatial-temporal feature, on the other hand, had more precise edges and shorter distances between classes. This comparison showed that the proposed feature did much better than the raw speech signal analysis regarding clarity and class distinction.

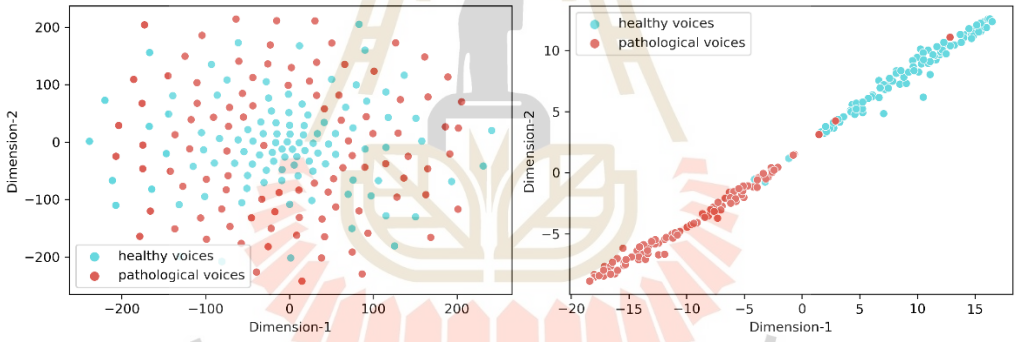


Figure 4.8 Views of t-SNE feature distributions (a) raw speech signals, (b) spatio-temporal features

Figure 4.8 shows the t-SNE analysis, which shows that the spatial-temporal features based on the RS-MSConvNet model are a better way to find voices that are not normal. The transparent edges and clear separation between classes suggest that this feature extraction method can make voice pathology diagnosis much more accurate.

4.9 Performance analysis of RS-MSConvNet models

This section gives a full breakdown of how well the RS-MSConvNet model works. This model was carefully created to find voices that are not normal. In this section, two different model versions were looked at: RS-MSConvNet-SVM and RS-MSConvNet-SVM with Feature Selection. Each of these models aims to improve the accuracy and reliability of pathological voice detection even more by using its unique architectural design and methods. The TORGO database was chosen to ensure the models would be evaluated fairly and thoroughly (Rudzicz,2008). This proves that the results can be trusted and used in other situations. The main goal is to thoroughly test and comprehend the detection abilities of the RS-MSConvNet and its variations. This will help determine what works best and what needs fixing.

4.9.1 Assessing RS-MSConvNet-SVM

The SVM hyperparameter tuning is very complicated to get the best performance, especially when using the RS-MSConvNet-SVM and RS-MSConvNet-SVM with Feature Selection. Because of the variety of multispectral data from remote sensing and the complexity of the RS-MSConvNet-SVM architecture, it is imperative to know how different hyperparameters affect how well the model works. This helps to better comprehend the model's behavior and lets it make better choices during the model deployment phase. This study conducted experiments with various combinations of hyperparameters, focusing on important parameters like the regularization parameter, kernel type, and kernel parameters. Many optimization techniques were also used to improve the tuning process and ensure the results were correct.

The most important results of our hyperparameter tuning experiments are shown in Table 4.1. A comparison of the model's performance with different hyperparameter settings and information on the settings that led to the best results is given.

Table 4.1 SVM hyperparameter adjustment for the RS-MSConvNet-SVM model.

C	Gamma (γ)	Score
0.1	scale	0.916
	0.01	0.766
	0.05	0.859
	0.1	0.891
	0.5	0.909
	1	0.920
	5	0.933
	10	0.936
0.5	scale	0.923
	0.01	0.816
	0.05	0.893
	0.1	0.905
	0.5	0.919
	1	0.925
	5	0.938
	10	0.946
1	scale	0.926
	0.01	0.837
	0.05	0.901
	0.1	0.950
	0.5	0.945
	1	0.930
	5	0.923
	10	0.910
5	scale	0.875
	0.01	0.883
	0.05	0.913
	0.1	0.918

Table 4.1 SVM hyperparameter adjustment for the RS-MSConvNet-SVM model (continued).

C	Gamma (γ)	Score
5	0.5	0.873
	1	0.876
	5	0.888
	10	0.882
10	scale	0.875
	0.01	0.892
	0.05	0.917
	0.1	0.922
	0.5	0.873
	1	0.878
	5	0.885
	10	0.879

In Table 4.1, analysis of the RS-MSConvNet-SVM model shows how its settings for hyperparameters and how well it works are closely linked. The best configuration was found when C = 1 and γ = 0.1. It got a performance score of 0.950, beating all the other tested configurations. The parameters should work together to make the model as accurate as possible.

4.9.2 Enhancing RS-MSConvNet-SVM with PSO

This section improves the results of applying PSO for hyperparameter tuning in the RS-MSConvNet-SVM with PSO Feature Selection model (Shami, 2022). The experimental setup, choice of hyperparameters for tuning, and the subsequent results are presented in detail. Additionally, an analysis of the importance and impacts of the results is provided.

Table 4.2 PSO hyperparameter adjustment for the RS-MSConvNet-SVM with PSO feature selection model.

W	C_1	C_2	Score
0.1	1	1	79.17
		2	79.99
		3	79.58
	2	1	79.14
		2	81.17
		3	80.22
	3	1	81.14
		2	80.39
		3	80.60
0.2	1	1	78.43
		2	79.95
		3	79.27
	2	1	78.43
		2	81.45
		3	80.26
	3	1	80.16
		2	80.22
		3	80.60
0.3	1	1	80.26
		2	78.56
		3	80.60
	2	1	79.44
		2	81.04
		3	79.24
	3	1	80.63
		2	79.85
		3	79.58

Table 4.2 PSO hyperparameter adjustment for the RS-MSConvNet-SVM with PSO feature selection model (continued).

W	C_1	C_2	Score
0.4	1	1	80.22
		2	79.44
		3	80.12
	2	1	79.65
		2	79.14
		3	81.07
	3	1	80.09
		2	79.58
		3	81.41
0.5	1	1	79.48
		2	79.04
		3	79.27
	2	1	79.85
		2	81.07
		3	80.29
	3	1	80.50
		2	80.73
		3	81.04
0.6	1	1	79.27
		2	79.55
		3	79.85
	2	1	80.36
		2	79.65
		3	79.75
	3	1	79.85
		2	80.12
		3	80.46

Table 4.2 PSO hyperparameter adjustment for the RS-MSConvNet-SVM with PSO feature selection model (continued).

W	C_1	C_2	Score
0.7	1	1	78.80
		2	79.75
		3	81.48
	2	1	80.16
		2	79.75
		3	80.63
	3	1	81.11
		2	81.21
		3	79.95
0.8	1	1	80.43
		2	80.33
		3	79.55
	2	1	79.72
		2	79.14
		3	79.99
	3	1	80.70
		2	80.53
		3	79.92
0.9	1	1	79.88
		2	80.05
		3	80.05
	2	1	80.39
		2	80.43
		3	80.02
	3	1	80.16
		2	80.22
		3	81.00

Table 4.2 PSO hyperparameter adjustment for the RS-MSConvNet-SVM with PSO feature selection model (continued).

W	C_1	C_2	Score
1	1	1	80.16
		2	79.41
		3	79.21
	2	1	80.36
		2	80.46
		3	79.72
	3	1	81.00
		2	79.61
		3	79.78
2	1	1	80.90
		2	79.61
		3	79.38
	2	1	80.29
		2	79.75
		3	79.41
	3	1	79.92
		2	81.31
		3	80.70
3	1	1	78.73
		2	79.10
		3	80.29
	2	1	78.43
		2	79.72
		3	79.27
	3	1	79.00
		2	80.80
		3	81.00

Table 4.2 PSO hyperparameter adjustment for the RS-MSConvNet-SVM with PSO feature selection model (continued).

W	C_1	C_2	Score
4	1	1	78.90
		2	80.26
		3	80.22
	2	1	78.56
		2	78.77
		3	79.51
	3	1	78.09
		2	79.82
		3	79.78
5	1	1	78.56
		2	79.48
		3	78.70
	2	1	78.63
		2	78.63
		3	79.34
	3	1	78.43
		2	78.63
		3	80.09

In Table 4.2, the PSO hyperparameters W, C_1, and C_2 settings significantly affect how well the RS-MSConvNet-SVM with PSO Feature Selection model works. These hyperparameters are very important for fine-tuning the model to perform at its best. The combination of W = 0.7, C_1 = 1, and C_2 = 3 stood out from the others tested because it got the best score of 81.48. This shows the importance of carefully changing the hyperparameters to get the most out of the RS-MSConvNet-SVM with PSO Feature Selection model.

4.9.3 Comparing RS-MSConvNet model variants

In the final analysis, raw speech data are used to test how well different proposed models can tell the difference between healthy and unhealthy voices. The three versions of the RS-MSConvNet model are looked at closely and judged on their accuracy, sensitivity, and specificity. The RS-MSConvNet-SVM model uses a Support Vector Machine to clarify decision-making processes. RS-MSConvNet-SVM with PSO Feature Selection, on the other hand, improves the current model by adding Particle Swarm Optimization to make classification work better. All models were tested using the same methods, ensuring that comparing their voice data classification abilities was fair and transparent. The full results can be found in Table 4.3.

Table. 4.3 Comparison performance of the proposed model

Classifier	Input	Accuracy (%)	Sensitivity (%)	Specificity (%)
RS-MSConvNet	Raw speech	86.46	83.04	87.88
RS-MSConvNet-SVM	Raw speech	87.61	78.86	91.23
RS-MSConvNet-SVM with PSO	Raw speech	88.09	80.49	91.23

Table 4.3 shows that out of the three models offered, the RS-MSConvNet-SVM with PSO Feature Selection model is the most accurate overall. In other words, this model would be the best choice for tasks or uses where getting the highest percentage of correct classifications is very important. But subtleties start to show up by looking more closely at the details of performance metrics. Sensitivity, called the True Positive Rate, measures how well a model can find positive samples. Sensitivity becomes an important metric when missing positive classifications can have significant effects, like when trying to diagnose a medical condition, and not seeing the need can be harmful. The RS-MSConvNet model does better in this situation than the others, though only

by a small amount. It has shown that it can find positive samples more accurately than the other two models.

On the other hand, specificity is another important metric that shows how well the model can classify negative samples. This metric is vital when false positives can cause actions or costs that are not needed. For example, if a screening test has a lot of false positives, it could lead to more tests or even treatments that are not required. Regarding specificity, both the RS-MSConvNet-SVM and RS-MSConvNet-SVM with PSO Feature Selection models do a great job. Since avoiding false positives is very important in these situations, either of these models might be the best choice.

4.10 Summary

This chapter's comprehensive analysis of the RS-MSConvNet models, specifically the RS-MSConvNet-SVM and its variant with Feature Selection, underscores the significance of meticulous hyperparameter tuning and architectural design in enhancing pathological voice detection. The study reveals that optimal performance is achieved through precise adjustments in model parameters, with the RS-MSConvNet-SVM model attaining its highest efficiency at specific settings. Integrating Particle Swarm Optimization in the RS-MSConvNet-SVM with the Feature Selection model further elevates its accuracy, showcasing the effectiveness of this approach. Comparative evaluation of the different RS-MSConvNet models indicates the superior accuracy of the RS-MSConvNet-SVM with PSO Feature Selection model while also highlighting each model's nuanced strengths in sensitivity and specificity. This research demonstrates the advanced capabilities of these models in voice pathology detection and provides valuable insights into the critical aspects of model optimization.

CHAPTER V
CONCLUSIONS

5.1 Conclusions

Voice healthcare is constantly changing, but one of the most essential things that can be done immediately to help people is finding voices that are not working directly. The idea for the RS-MSConvNet came from this thesis, which went into great detail about how to find different voices. This new design includes a multi-scale convolution neural network, spatial-temporal features, and an FC layer for sorting things into groups. Different fixed-length segments were looked at until the best one, 500 ms, was found. This was very important for the model's performance. By creating the RS-MSConvNet-SVM hybrid model, The RS-MSConvNet made progress. The RS-MSConvNet's trainable feature representation in this model is combined with an SVM's strong classification abilities. Testing the suggested models on the TORGO database revealed that they worked well: the RS-MSConvNet model achieved an impressive 86.46% accuracy, which was higher than other baseline systems; the hybrid RS-MSConvNet-SVM with PSO feature selection model did even better, reaching 87.61%.

This thesis adds a new structure to finding voices that are not normal and shows how essential hybrid models are for improving performance. These results indicate that neural networks and regular machine learning models can work together, which opens the door for new voice detection methods in the future. These models could be used in many ways in voice clinics and telemonitoring systems in the real world. The potential applications of these models in real-world voice clinics and telemonitoring systems are vast, promising transformative changes in the early detection and management of voice-related pathologies.

5.2 Future works

The RS-MSConvNet and hybrid RS-MSConvNet-SVM models, which are better at finding pathological voices and have added PSO feature selection, have set new standards in the field. The accuracy of these models shows how useful they could be, but deep learning models can continually be improved. To make models even more reliable, this includes improving neural network layers and using a variety of activation functions to make the models work better. It is also essential to do more research that uses datasets other than the TORGO database to see how well the models work with people who speak different languages and belong to various groups.

The demand for real-time pathological voice detection is growing. This shows how important it is to use these models in real-life situations. Also, combining these models with more advanced diagnostic methods, like medical imaging, could create a more complete diagnostic platform that would make it easier to find voice pathologies. Making these models easy to understand is essential, especially regarding healthcare decisions that have significant effects. Including more types of voice and speech disorders in research could make these studies more useful. Lastly, to make these models work in real life, more work must be done to make them work with medical equipment and in places with few resources. Models that work well on low-power devices should be given priority.

5.3 Thesis suggestions

Deep learning has changed healthcare in the past few years. Researchers are very excited that pathological voice detection is a promising new field. The main goal of this thesis is to create more advanced neural network architectures. The goal is to make models that can effectively and accurately detect pathological voice problems. These are thesis suggestions. Advanced architectural enhancements for pathological voice detection: Dive deeper into the intricacies of neural network architectures to

develop more advanced and efficient models for pathological voice detection. Explore novel neural network structures, attention mechanisms, and more.

1) Voice and medical imaging data for mixed evaluation: Combining RS-MSConvNet features with medical imaging data creates a complete diagnostic platform. Analyze the advantages and disadvantages of using speech and video data to assist doctors in their medical choices.
2) Real-time deployment and edge computing for voice diagnostics: Look into the difficulties of using voice detection models in real-time situations, especially on edge devices. Find out how model complexity, accuracy, and computational limits affect each other.
3) Deep learning that can be understood in healthcare: Study and develop methods to make healthcare deep learning models without compromising accuracy.
4) Diagnostics for voice and speech disorders: The study should examine a broader range of voice and speech disorders. Learn about the problems and possible solutions for finding and diagnosing various diseases.
5) Developing hardware and software for voice detection models: Because of the need for practicality, it is essential to look into how developing hardware and software can improve voice detection models' performance, especially when resources are limited.

REFERENCES

Alhussein, M., & Muhammad, G. (2018). Voice pathology detection using deep learning on mobile healthcare framework. IEEE Access, 6, 41034-41041.

Ali, Z., Alsulaiman, M., Elamvazuthi, I., Muhammad, G., Mesallam, T. A., Farahat, M., & Malki, K. H. (2016). Voice pathology detection based on the modified voice contour and SVM. Biologically Inspired Cognitive Architectures, 15, 10-18.

Al-Nasheri, A., Muhammad, G., Alsulaiman, M., Ali, Z., Malki, K. H., Mesallam, T. A., & Ibrahim, M. F. (2017). Voice pathology detection and classification using auto-correlation and entropy features in different frequency regions. IEEE Access, 6, 6961-6974.

Arjmandi, M. K., & Pooyan, M. (2012). An optimum algorithm in pathological voice quality assessment using wavelet-packet-based features, linear discriminant analysis and support vector machine. Biomedical signal processing and control, 7(1), 3-19.

Bačanin Džakula, N. (2019). Convolutional neural network layers and architectures. In Sinteza 2019-International Scientific Conference on Information Technology and Data Related Research (pp. 445-451). Singidunum University.

Barreira, R. R., & Ling, L. L. (2020). Kullback–Leibler divergence and sample skewness for pathological voice quality assessment. Biomedical Signal Processing and Control, 57, 101697.

Birjandtalab, J., Pouyan, M. B., & Nourani, M. (2016, February). Nonlinear dimension reduction for EEG-based epileptic seizure detection. In 2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI) (pp. 595-598). IEEE.

.

Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer.

Bouchard, L. (2022). Introduction to Convolutional Neural Networks (CNNs): The Most Popular Deep Learning Architecture. Medium. Retrieved from https://medium.com/what-is-artificial-intelligence/introduction-to-convolutional-neural-networks-cnns-the-most-popular-deep-learning-architecture-b938f62f133f

Celebi, M. E., & Aydin, K. (Eds.). (2016). Unsupervised learning algorithms (Vol. 9, p. 103). Cham: Springer.

De, S., Yadav, A., Jacobs, D., & Goldstein, T. (2016). Big batch SGD: Automated inference using adaptive batch sizes. arXiv preprint arXiv:1610.05792.

Dhillon, A., & Verma, G. K. (2020). Convolutional neural network: a review of models, methodologies and applications to object detection. Progress in Artificial Intelligence, 9(2), 85-112.

Dhuma, B. (Year). Which pooling method is better: MaxPooling vs MinPooling vs Average Pooling. Medium. https://medium.com/@bdhuma/which-pooling-method-is-better-maxpooling-vs-minpooling-vs-average-pooling-95fb03f45a9

Doshi, R., Chen, Y., Jiang, L., Zhang, X., Biadsy, F., Ramabhadran, B., ... & Moreno, P. J. (2021, June). Extending parrotron: An end-to-end, speech conversion and speech recognition model for atypical speech. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6988-6992). IEEE.

Eberhart, Russell, and James Kennedy. "A new optimizer using particle swarm theory." MHS'95. Proceedings of the sixth international symposium on micro machine and human science. Ieee, 1995.

Edureka. (2022). AI vs Machine Learning vs Deep Learning. Retrieved from https://www.edureka.co/blog/ai-vs-machine-learning-vs-deep-learning/

El Emary, I. M. M., Fezari, M., & Amara, F. (2014, November). Towards developing a voice pathologies detection system. Journal of Communication Technology and Electronics, 59(11), 1280–1288.

Engelbrecht, Andries P. Computational intelligence: an introduction. John Wiley & Sons, 2007.

Fang, S. H., Tsao, Y., Hsiao, M. J., Chen, J. Y., Lai, Y. H., Lin, F. C., & Wang, C. T. (2019). Detection of pathological voice using cepstrum vectors: A deep learning approach. Journal of Voice, 33(5), 634-641.

Fayek, H. M., Lech, M., & Cavedon, L. (2017). Evaluating deep learning architectures for speech emotion recognition. Neural Networks, 92, 60-68.

Gashi, I., Stankovic, V., Leita, C., & Thonnard, O. (2009, July). An experimental study of diversity with off-the-shelf antivirus engines. In 2009 Eighth IEEE International Symposium on Network Computing and Applications (pp. 4-11). IEEE.

Gómez-Vilda, P., Fernández-Baillo, R., Rodellar-Biarge, V., Lluis, V. N., Álvarez-Marquina, A., Mazaira-Fernández, L. M., ... & Godino-Llorente, J. I. (2009). Glottal source biometrical signature for voice pathology detection. Speech Communication, 51(9), 759-781.

Hamel, P., & Eck, D. (2010, August). Learning features from music audio with deep belief networks. In ISMIR (Vol. 10, pp. 339-344).

Harar, P., Alonso-Hernandezy, J. B., Mekyska, J., Galaz, Z., Burget, R., & Smekal, Z. (2017, July). Voice pathology detection using deep learning: a preliminary study. In 2017 international conference and workshop on bioinspired intelligence (IWOBI) (pp. 1-4). IEEE.

Hemmerling, D., Orozco-Arroyave, J. R., Skalski, A., Gajda, J., & Nöth, E. (2016, September). Automatic Detection of Parkinson's Disease Based on Modulated Vowels. In INTERSPEECH (pp. 1190-1194).

Hinton, G. E., & Roweis, S. (2002). Stochastic neighbor embedding. Advances in neural information processing systems, 15.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in science & engineering, 9(03), 90-95.

Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). pmlr.

Jamieson, A. R., Giger, M. L., Drukker, K., Li, H., Yuan, Y., & Bhooshan, N. (2010). Exploring nonlinear feature space dimension reduction and data representation in breast CADx with Laplacian eigenmaps and-SNE. Medical physics, 37(1), 339-351.

Janbakhshi, P., Kodrasi, I., & Bourlard, H. (2021, June). Automatic dysarthric speech detection exploiting pairwise distance-based convolutional neural networks. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 7328-7332). IEEE.

Kadiri, S. R., & Alku, P. (2019). Analysis and detection of pathological voice using glottal source features. IEEE Journal of Selected Topics in Signal Processing, 14(2), 367-379.

Kennedy, James. "The particle swarm: social adaptation of knowledge." Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97). IEEE, 1997.

Khairandish, M. O., Sharma, M., Jain, V., Chatterjee, J. M., & Jhanjhi, N. Z. (2022). A hybrid CNN-SVM threshold segmentation approach for tumor detection and classification of MRI brain images. Irbm, 43(4), 290-299.

Ko, W., Jeon, E., Jeong, S., & Suk, H. I. (2021). Multi-scale neural network for EEG representation learning in BCI. IEEE Computational Intelligence Magazine, 16(2), 31-45.

Kourkounakis, T., Hajavi, A., & Etemad, A. (2021). Fluentnet: End-to-end detection of stuttered speech disfluencies with deep learning. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, 2986-2999.

Koushik, J. (2016). Understanding convolutional neural networks. arXiv preprint arXiv:1605.09081.

LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10), 1995.

Li, Y., Liu, Y., Cui, W. G., Guo, Y. Z., Huang, H., & Hu, Z. Y. (2020). Epileptic seizure detection in EEG signals using a unified temporal-spectral squeeze-and-excitation network. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 28(4), 782-794.

Luque, A., Carrasco, A., Martín, A., & de Las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recognition, 91, 216-231.

Mallenahalli, N., & Sarma, T. H. (2018, July). A Tunable particle swarm size optimization algorithm for feature selection. In 2018 IEEE Congress on Evolutionary Computation (CEC) (pp. 1-7). IEEE.

Mesallam, T. A., Farahat, M., Malki, K. H., Alsulaiman, M., Ali, Z., Al-Nasheri, A., & Muhammad, G. (2017). Development of the arabic voice pathology database and its evaluation by using speech features and machine learning algorithms. Journal of healthcare engineering, 2017.

Narendra, N. P., & Alku, P. (2020). Glottal source information for pathological voice detection. IEEE Access, 8, 67745-67755.

Narendra, N. P., Schuller, B., & Alku, P. (2021). The detection of Parkinson's disease from speech using voice source information. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, 1925-1936.

Nasteski, V. (2017). An overview of the supervised machine learning methods. Horizons. b, 4, 51-62.

Phapatanaburi, K., Wang, L., Oo, Z., Li, W., Nakagawa, S., & Iwahashi, M. (2017). Noise robust voice activity detection using joint phase and magnitude based feature enhancement. Journal of ambient intelligence and humanized computing, 8, 845-859.

Poli, Riccardo. "Analysis of the publications on the applications of particle swarm optimisation." Journal of Artificial Evolution and Applications 2008 (2008): 1-10.

PyTorch. (2023). torch.nn.Conv2d. Retrieved from: https://pytorch.org/docs/stable/generated/torch.nn.Conv2d.html

Rios-Urrego, C. D., Moreno-Acevedo, S. A., Nöth, E., & Orozco-Arroyave, J. R. (2022, September). End-to-End Parkinson's Disease Detection Using a Deep Convolutional Recurrent Network. In International Conference on Text, Speech, and Dialogue (pp. 326-338). Cham: Springer International Publishing.

Ritchings, R. T., McGillion, M., & Moore, C. J. (2002). Pathological voice quality assessment using artificial neural networks. Medical engineering & physics, 24(7-8), 561-564.

Rudzicz, F., Van Lieshout, P., Hirst, G., Penn, G., Shein, F., & Wolff, T. (2008, December). Towards a comparative database of dysarthric articulation. In Proc. 8th Int. Seminar Speech Production (ISSP'08).

Russell, S. J., & Norvig, P. (2010). Artificial intelligence a modern approach. London.

Sabir, B., Rouda, F., Khazri, Y., Touri, B., & Moussetad, M. (2017). Improved algorithm for pathological and normal voices identification. International Journal of Electrical and Computer Engineering, 7(1), 238.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117.

Sellam, V., & Jagadeesan, J. (2014). Classification of normal and pathological voice using SVM and RBFNN. Journal of signal and Information Processing, 2014.

Shami, T. M., El-Saleh, A. A., Alswaitti, M., Al-Tashi, Q., Summakieh, M. A., & Mirjalili, S. (2022). Particle swarm optimization: A comprehensive survey. IEEE Access, 10, 10031-10061.

Shen, P., Changjun, Z., & Chen, X. (2011, August). Automatic speech emotion recognition using support vector machine. In Proceedings of 2011 international conference on electronic & mechanical engineering and information technology (Vol. 2, pp. 621-625). IEEE.

Shi, B. (2021). On the hyperparameters in stochastic gradient descent with momentum. arXiv preprint arXiv:2108.03947.

Shi, B., Su, W. J., & Jordan, M. I. (2020). On learning rates and schr\" odinger operators. arXiv preprint arXiv:2004.06977.

Shi, Yuhui, and Russell Eberhart. "A modified particle swarm optimizer." 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360). IEEE, 1998.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929-1958.

Teixeira, J. P., Fernandes, P. O., & Alves, N. (2017). Vocal acoustic analysis–classification of dysphonic voices with artificial neural networks. Procedia computer science, 121, 19-26.

Thuwajit, P., Rangpong, P., Sawangjai, P., Autthasan, P., Chaisaen, R., Banluesombatkul, N., ... & Wilaiprasitporn, T. (2021). EEGWaveNet: Multiscale CNN-based spatiotemporal feature extraction for EEG seizure detection. IEEE Transactions on Industrial Informatics, 18(8), 5547-5557.

Tirronen, S., Kadiri, S. R., & Alku, P. (2022). The effect of the MFCCs frame length in automatic voice pathology detection. Journal of Voice.

Trinh, N., & Darragh, O. B. (2019). Pathological speech classification using a convolutional neural network.

Rudzicz, F., Namasivayam, A. K., & Wolff, T. (2012). The TORGO database of acoustic and articulatory speech from speakers with dysarthria. Language Resources and Evaluation, 46, 523-541.

Vaiciukynas, E., Verikas, A., Gelzinis, A., Bacauskiene, M., Kons, Z., Satt, A., & Hoory, R. (2014). Fusion of voice signal information for detection of mild laryngeal pathology. Applied Soft Computing, 18, 91-103.

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of machine learning research, 9(11).

Vidyasagar, M. (2023). A tutorial introduction to reinforcement learning. SICE Journal of Control, Measurement, and System Integration, 16(1), 172-191.

Wang, L. (Ed.). (2005). Support vector machines: theory and applications (Vol. 177). Springer Science & Business Media.

Wang, L., Phapatanaburi, K., Go, Z., Nakagawa, S., Iwahashi, M., & Dang, J. (2017, July). Phase aware deep neural network for noise robust voice activity detection. In 2017 IEEE International Conference on Multimedia and Expo (ICME) (pp. 1087-1092). IEEE.

You, K., Long, M., Wang, J., & Jordan, M. I. (2019). How does learning rate decay help modern neural networks?. arXiv preprint arXiv:1908.01878.

APPENDIX

LIST OF PUBLICATIONS

List of Publications

International Journal Paper

Pathonsuwan, W., Phapatanaburi, K., Buayai, P., Jumphoo, T., Anchuen, P., Uthansakul, M., & Uthansakul, P. (2022). RS-MSConvNet: A Novel End-to-End Pathological Voice Detection Model. IEEE Access, 10, 120450-120461.

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 September 2022, accepted 31 October 2022, date of publication 4 November 2022, date of current version 21 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3219606

RESEARCH ARTICLE

RS-MSConvNet: A Novel End-to-End Pathological Voice Detection Model

WONGSATHON PATHONSUWAN[1], KHOMDET PHAPATANABURI[2], PRAWIT BUAYAI[3], TALIT JUMPHOO[1], PATIKORN ANCHUEN[4], MONTHIPPA UTHANSAKUL[1], (Member, IEEE), AND PEERAPONG UTHANSAKUL[1], (Member, IEEE)

[1]School of Telecommunication Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand
[2]Department of Telecommunication Engineering, Faculty of Engineering and Technology, Rajamangala University of Technology Isan (RMUTI), Nakhon Ratchasima 30000, Thailand
[3]Graduate Faculty of Interdisciplinary Research, University of Yamanashi, Kofu 400-8511, Japan
[4]Navaminda Kasatriyadhiraj Royal Air Force Academy, Bangkok 10220, Thailand

Corresponding authors: Khomdet Phapatanaburi (khomdet.ph@rmuti.ac.th) and Peerapong Uthansakul (uthansakul@g.sut.ac.th)

This work was supported in part by the Suranaree University of Technology (SUT), in part by the Thailand Science Research and Innovation (TRSI), and in part by the National Science Research and Innovation Fund (NSRF) through NRIIS under Grant 160361.

ABSTRACT Recent studies have reported the success of multi-scale convolution neural network (MSConvNet) model for many classification applications due to its powerful ability of exploring multi-scale convolution block to extract multi-scale representations to make a detection. However, a new design based on MSConvNet for pathological voice detection has not been explored. In this paper, we propose RS-MSConvNet, a novel end-to-end MSConvNet model using raw speech for pathological voice detection. The main contribution of the proposed RS-MSConvNet method is to exploit the multi-scale convolution block, followed by spatial-temporal feature block, and fully connected layer as classification. In addition, to further improve accuracy performance, we propose a novel hybrid detection model by integrating the feature extraction ability of the RS-MSConvNet model and the classifier of support vector machine (SVM) method, called RS-MSConvNet-SVM model. The effectiveness of our proposed models is investigated using the TORGO database. The experimental results reveal that the RS-MSConvNet model outperforms other baseline methods in the speaker-independent task. Moreover and as compared to the RS-MSConvNet-SVM model, a further improved accuracy is obtained using the RS-MSConvNet-SVM model. These outcomes exhibit that our proposed models are useful for pathological voice detection.

INDEX TERMS Pathological voice detection, end-to-end architecture, multi-scale convolution, spatial-temporal feature, hybrid model.

I. INTRODUCTION

Pathological voice detection is a technique of determining pathological voice or healthy voice from a provided utterance signal. It plays an important role in voice healthcare systems [1] such as voice clinics [2] and telemonitoring application [3], [4], [5] because the detection of changed speech is a diagnostic tool to identify the onset of disabling physical symptoms [6], where the results are exploited to screen patients at risk of having certain diseases. Moreover, the pathological voice detection is an essential pre-processing step for automatic speaker recognition for dysphonic voice assessment [7] and dysarthric speech recognition [8]. In this study, we focuse on a pathological voice detection, which is a subject area of the pattern recognition task in the field of biomedical and health informatics.

Typical pathological voice detection system can be divided into two groups: traditional pipeline system [9] and modern end-to-end system [10]. In the earlier studies [11], the systems usually consist of the front-end feature extraction and the back-end classifier. Based on traditional pipeline systems, the handcrafted design feature extraction converts speech signal into parametric representation while the back-end classifier learn feature representation for predicting

The associate editor coordinating the review of this manuscript and approving it for publication was Emanuele Lattanzi.

pathological/healthy voice class. For modern end-to-end systems that can extract features without using the handcrafted design feature extraction, a deep learning-based classifier used for predicting target classes is learned using a raw speech or its spectrogram. A brief survey based on existing traditional pipeline systems and modern end-to-end systems approaches for pathological voice detection is reviewed below.

Based on the traditional pipeline approach, most existing studies in pathological voice detection have focused on exploring effective hand-crafted design feature extraction with effective classifiers. Researchers have introduced various feature extraction methods for pathological voice detection. Mel-frequency cepstral coefficients (MFCC), Linear Predictive Cepstral Coefficients (LPCC), Linear Prediction Coefficients (LPC), Multi-Dimensional Voice Program (MDVP)-based features were proposed in [12]. Harmonics-to-noise ratio [13], Jitter [13], Shimmer [14], Kullback-Leibler divergence (KLD) histogram [15] and KLD higher amplitude suppression spectrum [15] were proposed for pathological voice detection. Autocorrelation and entropy features in different frequency regions were proposed in [16]. In addition to using the above mentioned individual features, the openSMILE set or Glottal source set-based fusion feature were introduced in order to combine acoustic features with statistical function sets or combine frequency-domain glottal and time-domain glottal feature sets with statistical function sets, respectively. Moreover, the combination of the openSMILE set and Glottal source set-based feature fusion was introduced [17] to fuse two merits based on different features. For the classifier, support vector machine (SVM) have been utilized as popular classifier in most previous studies [18], [19], [20], [21] because it can provide promising result for pathological voice detection. In addition to using SVM, researchers have applied various classifiers such as artificial neural networks [22], [23], linear discriminant analysis [24], Gaussian Mixture Model [25], and decision trees [26]. In all these traditional pipeline approaches, the ability of detecting pathological speech from healthy speech is strongly dependent on the effectiveness of effective handcrafted design feature extraction. This suggests that the detection performance requires expert knowledge in speech processing to devise relevant features.

Regarding pathological voice detection using end-to-end systems, previous works [11], [27], [28] have shown that they do not require expert feature engineering because deep learning models can be trained using either raw speech signal or its spectrum. For example, the combinations of convolutional neural network and multilayer perceptron (CNN-MLP)/long short-term memory (CNN-LSTM) using raw speech signal were proposed in [17]. The results showed that the CNN-MLP and CNN-LSTM could provide good results for pathological voice detection. However, using raw speech signal with any modification was not efficient enough as an input for training the end-to-end model under small training data. To further improve the end-to-end CNN-MLP and CNN-LSTM, the authors of [11] proposed to use glottal flow signal to replace raw speech signal as the input. The results showed that end-to-end CNN-LSTM and CNN-LSTM using glottal flow signal performed better than conventional CNN-MLP and CNN-LSTM using raw speech signal. Even though the end-to-end CNN-MLP and CNN-LSTM using either raw speech or glottal source signals could provide encouraging results, there is still an open research subject to design a new end-to-end model for pathological voice detection.

In this paper, a new end-to-end multi-scale convolution neural network architecture using raw speech, RS-MSConvNet is proposed for pathological voice detection. The main idea of the proposed architecture is to exploit multi-scale convolution block to scale the input information into different scaled representation, followed by spatial-temporal feature block and fully connected (FC) layer as a classifier block. In addition, to further improve detection accuracy, we propose a hybrid of RS-MSConvNet and SVM (RS-MSConvNet-SVM) models. Here, SVM classifier was explored to learn the automatically extracted features derived from fully trained RS-MSConvNet model. RS-MSConvNet and RS-MSConvNet-SVM provide promising results for speaker-independent pathologcal voice detection.

The contributions of this article can be summarized as follows:

1) A novel end-to-end model architecture, RS-MSConvNet is proposed to learn raw speech. The proposed RS-MSConvNet architecture which is end-to-end does not require expert knowledge in feature engineering.
2) We investigate our model on TORGO dataset. Here, the proposed RS-MSConvNet model performs comparably to other baseline systems in a speaker-independent approach.
3) The RS-MSConvNet is also modified using SVM method to replace FC layer as classification to learn the automatically extracted features derived from fully trained RS-MSConvNet model. The modified RS-MSConvNet is referred to as a hybrid RS-MSConvNet-SVM model architecture. The RS-MSConvNet-SVM model provides improved accuracy result, compared to the RS-MSConvNet classifier.

The rest of this paper is organized as follows: Our proposed methods are introduced in Section II. Section III describes pathological voice detection setup including the details of the database, network training, baseline method, and experimental evaluation. In Section IV, the results and discussions are presented. Section V presents our conclusion.

II. PROPOSED METHOD

A. RS-MSConvNet

The proposed RS-MSConvNet framework for pathological voice detection consists of pre-processing block, multi-scale

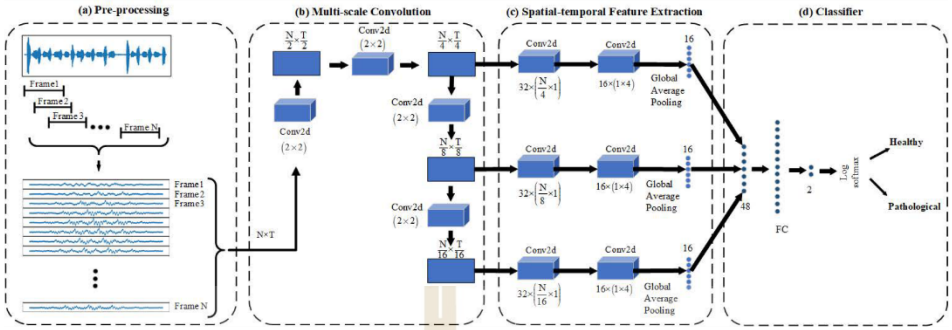


FIGURE 1. Overall visualization of the RS-MSConvNet architecture (a) Pre-processing (b) Multi-scale convolution block, (c) Spatial-temporal feature extraction block, and (d) Classifier block for two-class classification.

TABLE 1. Configuration of RS-MSConvNet architecture, where (*N*,*T*) are the dimension of input representation and k denotes the order of layer in block b.

Block	Layer	Kernel	Output	Activation	Parameters
b	Input		$(1, N, T)$		
	Conv2D	(2,2) stride = 2	$(1, \frac{N}{2}, \frac{T}{2})$	Linear	5
	Conv2D	(2,2) stride = 2	$(1, \frac{N}{4}, \frac{T}{4})$	Linear	5
	Conv2D	(2,2) stride = 2	$(1, \frac{N}{8}, \frac{T}{8})$	Linear	5
	Conv2D	(2,2) stride = 2	$(1, \frac{N}{16}, \frac{T}{16})$	Linear	5
c	Input		$(1, \frac{N}{2^k}, \frac{T}{2^k})$		
	Conv2D	$32x(\frac{N}{2^k},1)$	$(32, 1, \frac{T}{2^k})$		416
	Activation			Relu	
	Conv2D	16x(1,4)	$(16, 1, \frac{T}{2^k}-3)$		2064
	Activation			Relu	
	Activation			Dropout	
	Global average pooling		16		
d	Input		48		
	FC		128	Relu	
	FC		2	Linear	
	Classifier		1	Log Softmax	

convolution block, spatial-temporal feature extraction block, and classifier block as shown by the flowchart in Fig. 1. The configuration of the proposed RS-MSConvNet model is summarized in Table. 1.

1) PRE-PROCESSING BLOCK

This subsection describes how to form input data for training the RS-MSConvNet model. Pre-emphasis is initially employed to compensate the high-frequency component of the input speech signal. Next, the framing operations is used. In this paper, a 20 ms frame length and 10 ms frameshift of raw speech signals were divided into several speech frames and then a Hamming window is applied to enhance the harmonics and smooth the edges of the framed speech signals. Finally, a 2 dimension (2D) input data for training the proposed RS-MSConvNet model is formed by stacking the speech frames.

2) MULTI-SCALE CONVOLUTION BLOCK

Motivated by [29], [30], and [31], the feature pyramid networks based on Multi-scale convolution block has been proven to be effective feature technique in the built-up areas detection using synthetic aperture radar images [32] and electroencephalography seizure detection [33] because Multi-scale convolution block can extract multi-scale semantics information and make a more precise prediction by mean

of gathering more robust semantics information of scaled features. In this paper, Multi-scale convolution block was implemented to extract the 2D-input data into multi-scale features, making the designed detection model able to learn objects across a large range of scales. Each block was designed to extract the input fixed information in half the scale of the previous layer's resolution. The block could automatically learn the weight to effectively distinguish the valuable level features while reducing the signal to half.

In this block, the input data are shaped as (C, N, T) where C, N and T are defined as the number of channel, number of frames, and number of samples in each frame, respectively. Next, as seen in block (b), each 2D-convolution layer performs a convolution to reduce the input size with the same kernel size of (2, 2), a stride of 2, and no padding. By this configuration, the number of rows (frames) and columns (its samples) from k convolution layer become half of $k-1$ convolution layer providing the output size from the k layer to be $(1, \frac{N}{2^k}, \frac{T}{2^k})$. Here, outputs from the second to forth layers are used as the input for the next block for further extraction of spatial and temporal representation based on different field of views.

3) SPATIAL-TEMPORAL FEATURE EXTRACTION BLOCK

In this block, the objective is to extract spatial-temporal features from each output's scale of the Multi-scale convolution block. Here, two regular 2D-convolution layers are used to capture three last scaled output of the block. For the first 2D-convolution layers, three kernel sizes of $(\frac{N}{4}, 1)$, $(\frac{N}{8}, 1)$, $(\frac{N}{16}, 1)$ with a stride of 2, no padding, and output channels of 32 are used to capture the outputs from the second layer, third layer and forth layer of previous phase, respectively. For the second 2D-convolution layers, the same kernel size of (1,4) with a stride of 2, no padding, and channel outputs of 16 are used to extract the different output of the first layers. Finally, global average pooling is applied to the output passed from the two regular 2D-convolution layers. By this configuration, a total of 48 features (16 representations per each scale) are obtained to be the input for next block.

4) FULLY CONNECTED LAYER BLOCK

After two convolutional layers and global average pooling, the spatial feature module s' outputs of different scales are augmented and fed to FC layers. Log softmax is used as our last layer for predicting binary classes. Based on Log softmax, the logarithm of the prediction probability of binary classes is computed as follows:

$$\hat{y}(x_i) = \log\left(\frac{\exp(x_i)}{\sum_j \exp(x_j)}\right). \tag{1}$$

where x_i is the input vector with the i^{th} element and j is the number of classes (possible outcomes).

To calculate classification loss, cross entropy is implemented in this study. This loss function calculates the similarity between the label and predicted probability values as follows:

$$\mathcal{L}_{CE}(y, \hat{y}) = -\sum_{k=1}^{|class|} y_k \log \hat{y}_k. \tag{2}$$

where $\hat{y}$ and y are the predicted probability and the true label, respectively.

B. RS-MSConvNet-SVM

SVM-based method has been proven to be effective for both classification and regression problems, so it has been utilized for many applications [34] such as face recognition [35], electroencephalography seizure detection [36], and automatic emotion speech recognition [37]. Moreover, most previous studies have used SVM-based method as baseline classifier for pathological voice detection because it can deal with two-class classification problem. In this paper, the SVM-based method is applied to learn automatically optimized features based on the RS-MSConvNet model.

Motivated by [38] which integrated CNN as a trainable feature representation and SVM as a classifier, a hybrid CNN and SVM (CNN-SVM) provided better accuracy results than CNN model for tumor detection. This was attributed to the fact that the developed model combined the advantages of CNN and SVM models. Similarly, hybrid RS-MSConvNet and SVM (RS-MSConvNet-SVM) as shown in Fig. 2 is proposed by using SVM as the classification to replace the FC layers after the RS-MSConvNet classifier was fully trained as shown in Fig. 2 (a). In this paper, to construct the SVM in a hybrid model, we adopt radial basis function (RBF) and determined penalty parameter $\mathcal{C}$ and the optimal kernel parameter γ by investigating the validation data on the hybrid model learned by training data. Both training and validation data are explained in next section.

The implementation process of the RS-MSConvNet-SVM model is shown in Fig. 2 (b) and can be summarized as follows:

1) For the training process, the samples of training set were fed to RS-MSConvNet model.

2) After the RS-MSConvNet classifier was fully trained, the corresponding feature information could be automatically extracted for each input map.

3) The FC layers were replaced with SVM-based classifier to learn the automatically extracted feature vectors derived from the fully trained RS-MSConvNet classifier.

4) For the test process, the samples of test set were fed to the fully trained RS-MSConvNet classifier to obtain the automatically extracted features as the test feature representation.

5) The test feature data was fed to the well-trained SVM for predicting healthy or pathological class.

III. EXPERIMENTAL SETUP

A. DATABASE

TORGO [39] and UA-Speech [40] are commonly used databases for pathological voice detection. In this paper, TORGO database is used to investigate our RS-MSConvNet

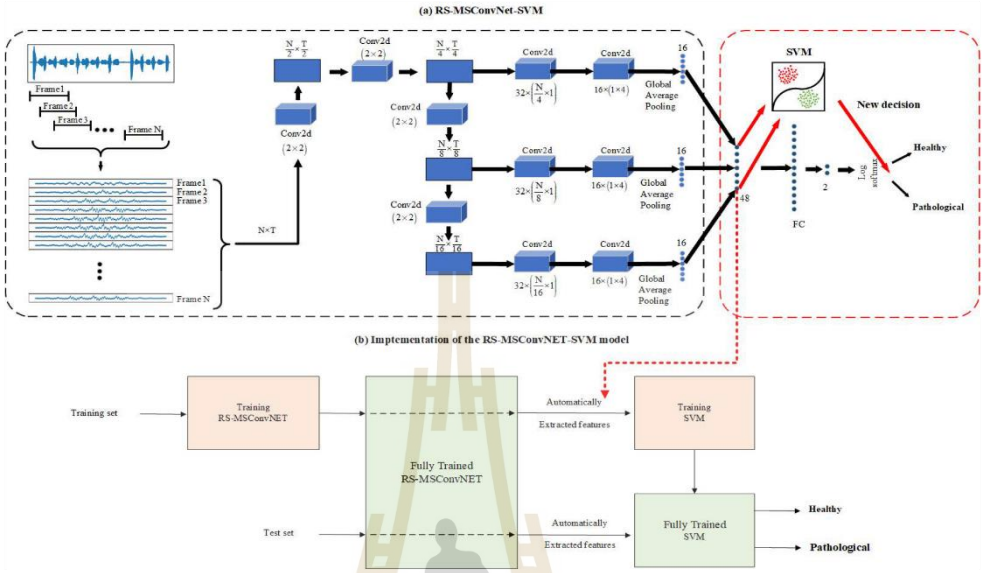


FIGURE 2. Implementation process of the RS-MSConvNET-SVM model.

TABLE 2. Details of the TORGO database.

Training	Validation	Testing
MC03, MC04 FC02, M02, M05, F01, F03	MC02, FC01, M01, M03	MC01, FC03, M04, F04

TABLE 3. Model parameters of the RS-MSConvNet and RS-MSConvNet-SVM models.

Parameters	RS-MSConvNet	RS-MSConvNet-SVM
Batch Size	256	256
Learning Rate	0.0001	0.0001
Dropout Rate	0.5	0.5
Epoch	1000	1000
SVM ($\mathcal{C}$)	N/A	1

and RS-MSConvNet-SVM models. The main reason for using this database is that the database is more challenging than UA-Speech database due to the limited data, which makes the end-to-end models based on the TORGO database difficult to achieve better accuracy than the end-to-end models based on the UA-Speech database as seen in [17]. Moreover, the results are also directly compared with the experimental settings as in [17]. The publicly available TORGO corpus was produced by three females (F01, F03, F04) with dysarthria, three healthy females, five males (M01, M02, M03, M04, M05) with dysarthria, and five healthy males (MC01, MC02, MC03, MC04). In this database, participants without dysarthria recorded approximately 900 utterances on average while participants with dysarthria recorded approximately 400 utterances on average. Further details of the TORGO database can be seen in [39]. All speech utterances were sampled at 16 kHz. In this study, since a substantial amount of silence was contained in the TORGO database, it need to be removed for training/testing classification model. To conduct the speaker-independent pathological voice detection as advised in [17] and [41], the database is divided into three sets: training subset (3,125 healthy and 1,491 pathological utterances with 3.5 hr), validation subset (944 healthy and 795 pathological utterances with 2 hr), and testing subset (2,087 healthy and 861 pathological utterances with 3 hr). Table. 2 summarizes all three subsets of TORGO database used for our experiments.

B. NETWORK TRAINING

In this paper, we used the PyTorch v1.10.1 framework to build the proposed method. NVIDIA RTX3090 with 24 GB memory was used to train the networks. Adam optimizer was exploited to optimize the loss function in each iteration of training process. The model parameters for training the RS-MSConvNet and RS-MSConvNet-SVM models are listed in Table. 3.

C. BASELINE SYSTEMS

Based on the same database and training/testing condition, the effectiveness of the proposed RS-MSConvNet

and RS-MSConvNet-SVM methods are compared with the results of five baseline system groups: OpenSMILE+SVM, Glottal+SVM, OpenSMILE-Glottal+SVM, and conventional end-to-end methods, modified end-to-end methods using glottal flow signal.

1) OpenSMILE+SVM methods: two acoustic feature extraction sets obtained using the OpenSMILE toolkit were used as the input feature information for the classifier. First OpenSMILE s' acoustic feature sets (OpenSMILE-1) with a total of 384 dimensions ((16 dimensions of the chosen acoustic features + 16 Δ dimensions) × 12 statistical functions) and second OpenSMILE s' acoustic feature sets (OpenSMILE-1) with a total of 6552 dimensions (56 dimensions of the chosen acoustic features + 56 Δ dimensions) × 39 statistical functions) were used as the input for the SVM classifier. The lists of OpenSMILE-1 set and OpenSMILE-2 set with its statistical information are summarized in Table. 4. For this baseline system, the SVM-based classifiers using the OpenSMILE-1 and OpenSMILE-2 sets are referred to as OpenSMILE-1+SVM and OpenSMILE-2+SVM, respectively.

2) Glottal+SVM methods: two glottal feature sets were used as the input feature for the SVM-based classifier. First glottal feature set (Glottal-1) with 192 feature vectors ((12 dimensions of the chosen acoustic features + 12 Δ dimensions) × 8 statistical functions) was obtained by capturing glottal flow signal using several time-and frequency-domain feature extraction method as listed in Table. 5. For second glottal feature set (Glottal-2), the principal component analysis (PCA) with 30 principal component weights was applied to the normalized Glottal-1 to calculate 480-dimension features ((30 dimensions of the chosen acoustic features + 30 Δ dimensions) × 8 statistical functions). For this baseline system, the SVM-based classifiers with the Glottal-1 and Glottal-2 sets are referred to as Glottal-1+SVM and Glottal-2+SVM, respectively.

3) OpenSMILE-Glottal+SVM methods: To take the advantages of two types of feature extraction sets mentioned above, the OpenSMILE-1/OpenSMILE-2 and Glottal-1/Glottal-2 sets were joined as the input for further improving the SVM-based classifier. Here, SVM using joint OpenSMILE-1 and Glottal-1 sets, joint OpenSMILE-1 and Glottal-2 sets, joint OpenSMILE-2 and Glottal-1 sets, and joint OpenSMILE-2 and Glottal-2 sets were referred to as OpenSMILE-1-Glottal-1+SVM, OpenSMILE-1-Glottal-2+SVM, OpenSMILE-1-Glottal-1+SVM, and OpenSMILE-1-Glottal-2+SVM, respectively.

4) Conventional end-to-end methods: the CNN-MLP and CNN-LSTM methods were used as baseline end-to-end method using raw speech.

5) Modified End-to-end methods using glottal flow: In similar way as conventional end-to-end methods, the CNN-MLP and CNN-LSTM methods were also used. Unlike conventional end-to-end methods, the glottal flow signals were used to replace raw speech signals as the input for CNN-MLP and CNN-LSTM-based classifier.

Further details of five baseline methods compared with our proposed methods can be seen in [17].

D. EXPERIMENTAL EVALUATION

In order to investigate the effectiveness of our proposed methods, three common evaluation criteria suggested in [42] are used: classification accuracy, sensitivity, and specificity. Accuracy is computed as:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}. \tag{3}$$

where TP and TN are the true pathological voice and true healthy voice where the fully trained network correctly predicts the pathological voice and healthy voice classes. FP and FN are the true pathological voice and true healthy voice which are incorrectly classified.

The sensitivity and specificity are calculated as follows:

$$Sensitivity = \frac{TP}{TP + FN}. \tag{4}$$

$$Specificity = \frac{TN}{TN + FP}. \tag{5}$$

IV. RESULTS AND DISCUSSIONS

A. RESULTS ON RS-MSConvNet

This subsection reports the performance of RS-MSConvNet. The following conclusions based on varying the configuration of parameters can be drawn:

- Since the fixed-length segments has an effect on the performance of end-to-end network, it is important to find out suitable fixed-length segments. In this paper, different fixed-length segments of 240-ms, 250-ms, 500-ms, 1 s and 3 s were first investigated to find out the optimal fixed-length segment. Table. 6 reports the results of RS-MSconvnet model using different fixed-length segments.

It can be seen that the fixed-length segment of 500 ms provided the best performance compared with the others. The reasons were that the network could not sufficiently learn the overly short data of raw speech signals (240 or 250 ms), which might have not contained enough pathological information while a fixed-length speech segment longer than 500 ms could not be exploited due to the short durations of some of the vowels as summarized in [47]. Moreover, the fixed-length segment longer than 500 ms might have led to overly small training data making the fully trained network incompetent to detect pathological speech from healthy speech. The results indicate that the fixed-length segment of 500 ms (49 × 320 pixels) was the most suitable for the RS-MSConvNet model.

- Many trials, which is not reported in this section (data not shown), were conducted by adding/reducing the convolution layers and changing the parameters in Multi-scale convolution block and spatial-temporal feature extraction block but they did not achieve better results. Moreover, batch normalization was also applied to spatial-temporal feature extraction block and it could not improve the detection performance. This outcome means that changing Multi-scale

TABLE 4. Two acoustic feature sets with their statistical functions computed with the openSMILE toolkit.

Feature sets	Acoustic features	Statistical functionals
openSMILE-1	zero-crossing rate, RMS-energy, pitch, MFCCs (12 coefficients), voicing probability	min (or max) value and its relative position, range, median,kurtosis, skewness, standard deviation, 2 linear regression coeff. and quadratic error
openSMILE-2	log-energy, zero-crossing rate, pitch, MFCCs (13 coefficients), Mel-spectrum (26 coefficients), jitter, shimmer, spectral flux, voicing probability, roll-off points, spectral centroid, position of spectral minimum and maximum	min (or max) value and its relative position, range, median, kurtosis, skewness, standard deviation, 2 linear regression coeff., linear and quadratic errors, 3 quartiles, 2 percentiles (95% & 98%), 3 inter-quartile errors, number of peaks, mean of peaks, mean distance between peaks, geometric, arithmetic and quadratic means

TABLE 5. Time-domain glottal features, frequency-domain glottal features, and statistical functions used for Glottal-1 feature set.

Time-domain glottal features	Frequency-domain glottal features	Statistical functionals
Amplitude quotient, Closing quotient, Speed quotient computed from the primary glottal opening, Speed quotient, computed from the secondary glottal opening, Normalized amplitude quotient, Open quotient, extracted from the LF model, Open quotient obtained from the primary glottal opening, and Open quotient obtained from the secondary glottal opening	Difference between the lowest two glottal harmonics, Harmonic richness factor, and Parabolic spectrum parameter	Skewness, Standard deviation, Kurtosis, Maximum, Minimum, Median, Mean, and Range

TABLE 6. Performance of the proposed RS-ConvNet classifier using different input speech lengths.

Input speech length	Accuracy(%)	Sensitivity(%)	Specificity(%)
240 ms	79.44	72.36	82.37
250 ms	82.19	71.89	86.44
500 ms	**86.46**	**83.03**	**87.88**
1s	82.87	82.22	83.13
3s	80.08	70.15	87.01

convolution block, spatial-temporal feature extraction block, and the parameter was unsuitable for our RS-MSConvNet method.

• Finally, since the number of FC layers has an effect on the detection performance of pathological voice, the numbers of FC layers was varied from 1 to 5 layers. Table. 3 shows the comparison among different FC layers. It was found that the detection performances decreased while using more than two layers. This is because one FC layer is suitable for the classification based on two classes and the limited training data suggested in [43] and [44]. This suggested that using one FC layer as classification was suitable for our RS-ConvNet model.

To visualize discriminating information using the scaled feature representation for pathological voice detection, a healthy voice and a pathological voice signal were chosen to be fed into a fully trained RS-ConvNet model. The output representation derived from second convolution layer to fourth layers were then compared to show discriminating feature information of healthy and pathological voices. In this paper, the representation images are displayed using the matplotlib function based on bilinear interpolation method [45]. Here, since the fixed-length segment of 500 ms which provided the best results mentioned above was used as the input for RS-MSConvNet, the output sizes of second layer, third layer, fourth layer are 10×80 pixels, 5×40 pixels, and 2×20 pixels, respectively. Fig. 4 shows a comparison of feature representation between healthy voice and a pathological voice signals with a similar amplitude signature. We can observe from Fig. 4 that the convolution layers provided different representation between healthy voice and pathological voice. This indicated that the proposed Multi-scale scale convolution block could give discriminative features for the pathological voice detection.

Next, to observe discriminating ability using the spatial-temporal features derived from the trained RS-MSConvNet

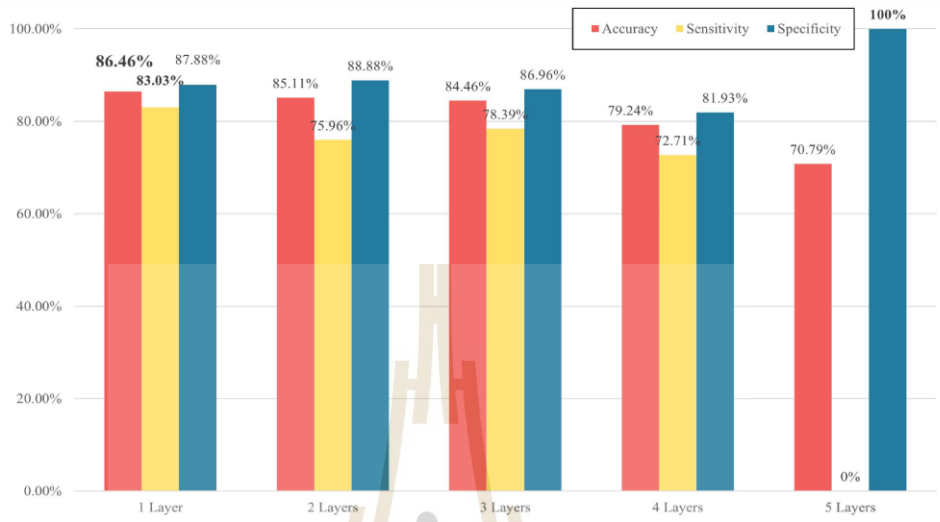


FIGURE 3. Performance of the proposed RS-MSConvNet classifier using different layers.

model for detecting pathological voices, the t-distributed stochastic neighbor embedding (t-SNE) [46] which is a commonly used method for dimensionality reduction was exploited to consider the distributions between healthy voice and pathological voice categories. Here, 200 pathological and 200 healthy voice samples were selected to show the distributions of the two classes based on the t-SNE analysis. Fig. 5 shows visual distribution of the spatial-temporal feature derived from the trained RS-MSConvNet model. As seen in Fig. 5 (a), the data distributions of the different classes using raw speech signals without any feature extraction were significantly overlapped. This caused difficulty in distinguishing the different voices. By comparing Fig. 5 (a) with (b), it can be seen that the data distribution of the proposed spatial-temporal feature performed better than using raw speech signals without any processing method because it provided clear contours and small inter-class distances. This suggested that the spatial-temporal feature based on the RS-MSConvNet could be useful for pathological voice detection.

B. RESULTS ON RS-MSConvNet-SVM

This subsection presents the results of RS-MSConvNet-SVM. Because the γ value directly affects the SVM-based detection performance, it is important to find out the optimal γ. Here, the γ values varied from 0.1 to 30 by the step size of 0.1, with the optimal γ at 0.1 which provided the highest accuracy by investigating the validation set on the hybrid model trained by training set. Therefore, the fully trained RS-MSConvNet model using the optimal γ at 0.1 was used to evaluate the testing set because the decision ranked by the distance between the RS-MSConvNet based-automatically extracted features and the hyperplane of the trained SVM model achieved the highest number of correct predictions. Fig. 6 shows the result of the RS-MSConvNet-SVM model compared with the results of the RS-MSConvNet model.

As seen in Fig.6, improved accuracy performance was obtained using the hybrid model. The accuracy was improved from the RS-MSConvNet with 86.46 % to the RS-MSConvNet-SVM with 87.61 %. This can be attributed to the decision ranked by the distance between the RS-MSConvNet based-automatically extracted features and the hyperplane of the trained SVM model performing higher specificity compared with the RS-MSConvNet classifier, which led to directly improving the detection accuracy. This result indicated that the RS-MSConvNet-SVM seems useful for detecting pathological voice from healthy voice.

C. COMPARISON WITH BASELINE SYSTEMS

In this subsection, the performances of our proposed methods are compared to those of some known systems. As mentioned in the introduction section, some systems may not be discussed due to the experiments being based on speaker-dependent approach and different database from our experiments. Here, the results based only on the TORGO database for a speaker-independent approach, which is the same condition as our experiments, were compared. Table. 7 shows the results of some known systems compared to our proposed methods.

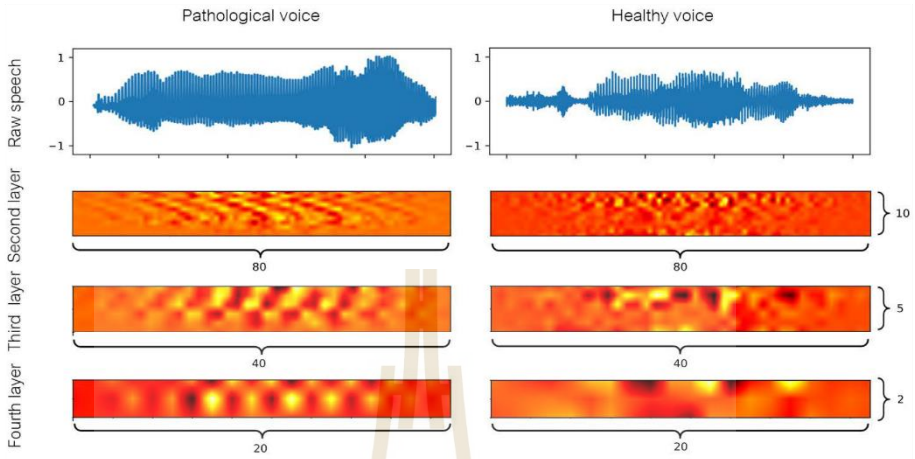


FIGURE 4. Comparison representation of raw speech and the outputs from second convolution layer to fourth convolution layer. Both are from patient #01 speaking "Zero".

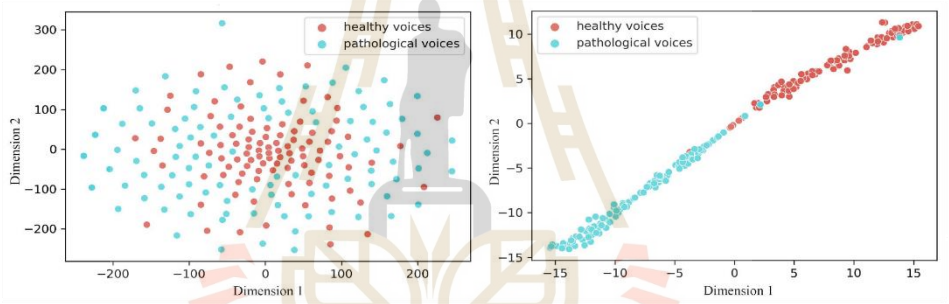


FIGURE 5. Visual distributions of the different features based on t-SNE. (a) raw speech signals, (b) spatio-temporal features.

TABLE 7. Comparison with baseline systems on the TORGO database.

Systems (results in [17])	Feature sets	Classifier	Accuracy (%)	Sensitivity (%)	Specificity (%)
Conventional pipeline	openSMILE-1	SVM	78.24	72.94	83.54
	openSMILE-2	SVM	80.62	73.73	87.52
	Glottal-1	SVM	67.17	71.22	63.12
	Glottal-2	SVM	66.93	71.55	62.17
	openSMILE-1 + Glottal-1	SVM	79.62	73.21	86.03
	openSMILE-2 + Glottal-1	SVM	**82.12**	**79.02**	85.22
	openSMILE-1 + Glottal-2	SVM	80.63	72.59	**88.68**
	openSMILE-2 + Glottal-2	SVM	81.35	76.83	85.87
Systems (results in [17])	Input	Classifier	Accuracy (%)	Sensitivity (%)	Specificity (%)
Conventional end-to-end	Raw speech	CNN-MLP	78.83	82.85	**76.24**
	Glottal flow	CNN-MLP	**81.12**	**85.88**	75.26
	Raw speech	CNN-LSTM	71.15	78.45	66.17
	Glottal flow	CNN-LSTM	75.41	81.32	69.68
Our proposed	Raw speech	RS-MSConvNet-FC	86.46	**83.04**	87.88
	Raw speech	RS-MSConvNet-SVM ($\gamma = 0.1$)	**87.61**	78.86	**91.23**

As seen in Table. 7, the results obtained with the RS-MSConvNet and RS-MSConvNe models outperformed all known systems in terms of accuracy and specificity performances. For the specificity result, it was observed the end to end system approach (CNN-MLP) using glottal flow information performed better than the proposed methods because the glottal flow signal gave better discriminative information than raw speech signal as summarized in [47]

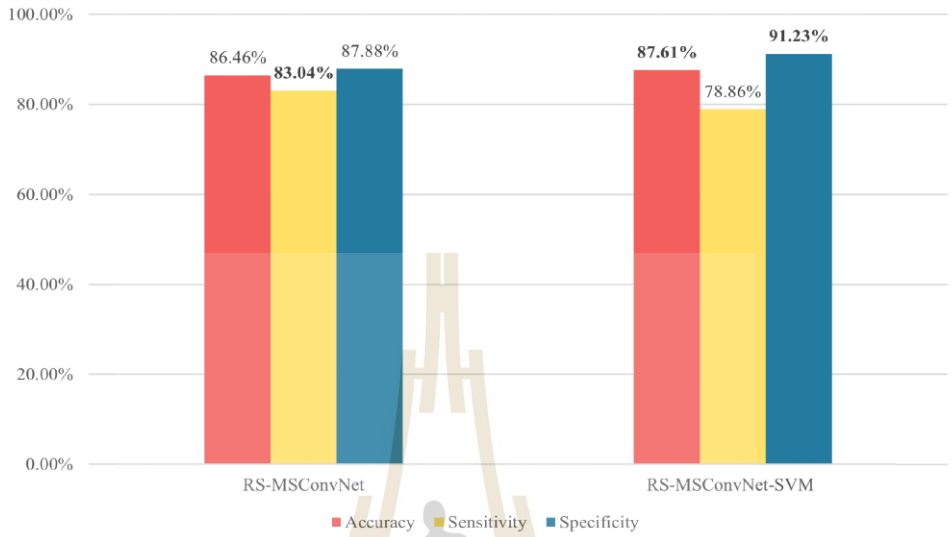


FIGURE 6. Comparison performance of the proposed RS-MSConvNet-SVM classifier and RS-MSConvNet classifier.

making the detection of pathological voice more specific. However, accuracy and specificity of the CNN-MLP using glottal flow information was worse than those of the proposed systems. This indicated that the proposed methods could give more reliable classification performance without requiring expert knowledge in pre-processing for computing alternative efficient signal to replace raw speech signal.

V. CONCLUSION

In this paper, we proposed a new RS-MSConvNet architecture for pathological voice detection. The main contribution of the proposed RS-MSConvNet method is to use Multi-scale convolution neural network, followed by spatial-temporal feature, and FC layer as classification. In addition, we proposed a hybrid model by integrating RS-MSConvNet as trainable feature presentation and support vector machine (SVM) as a classifier and referred to it as RS-MSConvNet-SVM model. The performances of our proposed models were evaluated using the TORGO database. From the experimental results, it was observed that the RS-MSConvNet gave the discriminating feature information between healthy and pathological voice via the t-SNE method and provided an accuracy of 86.46 %, which outperformed other baseline systems. In addition, improved accuracy performance was obtained using RS-MSConvNet-SVM model.The accuracy was improved from the RS-MSConvNet with 86.46 % to the RS-MSConvNet with 87.61 %. The results indicated that our proposed RS-MSConvNet and RS-MSConvNet-SVM approaches could be useful for pathological voice detection.

In the future, the effectiveness of using the attention mechanism will be explored to further improve our proposed RS-MSConvNet and RS-MSConvNet-SVM approaches. We will also use Glottal flow signal [48] to replace raw speech signal as the input of our proposed methods.

REFERENCES

[1] Y. Wu, C. Zhou, Z. Fan, D. Wu, X. Zhang, and Z. Tao, "Investigation and evaluation of glottal flow waveform for voice pathology detection," *IEEE Access*, vol. 9, pp. 30–44, 2021.

[2] C. Watters, B. Miller, M. Kelly, V. Burnay, Y. Karagama, and E. Chevretton, "Virtual voice clinics in the COVID-19 era: Have they been helpful?" *Eur. Arch. Oto-Rhino-Laryngology*, vol. 278, no. 10, pp. 4113–4118, Oct. 2021.

[3] M. Little, P. McSharry, E. Hunter, J. Spielman, and L. Ramig, "Suitability of dysphonia measurements for telemonitoring of Parkinson's disease," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 4, pp. 1010–1022, Apr. 2009.

[4] C. G. Goetz, G. T. Stebbins, D. Wolff, W. DeLeeuw, H. Bronte-Stewart, R. Elble, M. Hallett, J. Nutt, L. Ramig, T. Sanger, A. D. Wu, P. H. Kraus, L. M. Biasucci, E. A. Shamim, K. D. Sethi, J. Spielman, K. Kubota, A. S. Grove, E. Dishman, and C. B. Taylor, "Testing objective measures of motor impairment in early Parkinson's disease: Feasibility study of an at-home testing device," *Movement Disorders*, vol. 24, no. 4, pp. 551–556, Mar. 2009.

[5] P. Klumpp, T. Janu, T. Arias-Vergara, J. C. Vásquez-Correa, J. R. Orozco-Arroyave, and E. Nöth, "Apkinson—A mobile monitoring solution for Parkinson's disease," in *Proc. Interspeech*, Aug. 2017, pp. 1839–1843.

[6] A. E. Aronson and D. M. Bless, *Clinical Voice Disorders*. New York, NY, USA: Springer, 2009.

[7] C. Fredouille, G. Pouchoulin, J.-F. Bonastre, M. Azzarello, A. Giovanni, and A. Ghio, "Application of automatic speaker recognition techniques to pathological voice assessment (dysphonia)," in *Proc. Interspeech*, Sep. 2005, pp. 149–152.

[8] Y. Lin, L. Wang, S. Li, J. Dang, and C. Ding, "Staged knowledge distillation for End-to-End dysarthric speech recognition and speech attribute transcription," in *Proc. Interspeech*, Oct. 2020, pp. 4791–4795.

[9] N. P. Narendra, B. Schuller, and P. Alku, "The detection of Parkinson's disease from speech using voice source information," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 1925–1993, 2021.
[10] M. Alhussein and G. Muhammad, "Voice pathology detection using deep learning on mobile healthcare framework," *IEEE Access*, vol. 6, pp. 41034–41041, 2018.
[11] 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 *Proc. Int. Conf. Workshop Bioinspired Intell. (IWOBI)*, Jul. 2017, pp. 1–4.
[12] T. A. Mesallam, M. Farahat, K. H. Malki, M. Alsulaiman, Z. Ali, A. Al-Nasheri, and G. Muhammad, "Development of the Arabic voice pathology database and its evaluation by using speech features and machine learning algorithms," *J. Healthcare Eng.*, vol. 2017, pp. 1–13, Feb. 2017.
[13] B. Sabir, F. Rouda, Y. Khazri, B. Touri, and M. Moussetad, "Improved algorithm for pathological and normal voices identification," *Int. J. Elect. Comput. Eng.*, vol. 7, no. 1, pp. 238–243, 2017.
[14] E. Vaiciukynas, A. Verikas, A. Gelzinis, M. Bacauskiene, Z. Kons, A. Satt, and R. Hoory, "Fusion of voice signal information for detection of mild laryngeal pathology," *Appl. Soft Comput.*, vol. 18, pp. 91–103, May 2014.
[15] R. R. A. Barreira and L. L. Ling, "Kullback–Leibler divergence and sample skewness for pathological voice quality assessment," *Biomed. Signal Process. Control*, vol. 57, Mar. 2020, Art. no. 101697.
[16] A. Al-Nasheri, "Voice pathology detection and classification using autocorrelation and entropy features in different frequency regions," *IEEE Access*, vol. 6, pp. 6961–6974, 2018.
[17] N. P. Narendra and P. Alku, "Glottal source information for pathological voice detection," *IEEE Access*, vol. 8, pp. 67745–67755, 2020.
[18] M. K. Arjmandi and M. Pooyan, "An optimum algorithm in pathological voice quality assessment using wavelet-packet-based features, linear discriminant analysis and support vector machine," *Biomed. Signal Process. Control*, vol. 7, no. 1, pp. 3–19, Jan. 2012.
[19] V. Sellam and J. Jagadeesan, "Classification of normal and pathological voice using SVM and RBFNN," *J. Signal Inf. Process.*, vol. 5, no. 1, pp. 1–7, 2014.
[20] Z. Ali, "Voice pathology detection based on the modified voice contour and SVM," *Biol. Inspired Cognit. Archit.*, vol. 15, pp. 10–18, Jan. 2016.
[21] S.-H. Fang, Y. Tsao, M.-J. Hsiao, J.-Y. Chen, Y.-H. Lai, F.-C. Lin, and C.-T. Wang, "Detection of pathological voice using cepstrum vectors: A deep learning approach," *J. Voice*, vol. 33, no. 5, pp. 634–641, Sep. 2019.
[22] R. T. Ritchings, M. McGillion, and C. J. Moore, "Pathological voice quality assessment using artificial neural networks," *Med. Eng. Phys.*, vol. 24, nos. 7–8, pp. 561–564, Sep. 2002.
[23] J. P. Teixeira, P. O. Fernandes, and N. Alves, "Vocal acoustic analysis—Classification of dysphonic voices with artificial neural networks," *Proc. Comput. Sci.*, vol. 121, pp. 19–26, Dec. 2017.
[24] P. Gómez-Vilda, R. Fernández-Baillo, V. Rodellar-Biarge, V. N. Lluis, A. Álvarez-Marquina, L. M. Mazaira-Fernández, R. Martínez-Olalla, and J. I. Godino-Llorente, "Glottal source biometrical signature for voice pathology detection," *Speech Commun.*, vol. 51, no. 9, pp. 759–781, Sep. 2009.
[25] I. M. M. El Emary, M. Fezari, and F. Amara, "Towards developing a voice pathologies detection system," *J. Commun. Technol. Electron.*, vol. 59, no. 11, pp. 1280–1288, Nov. 2014.
[26] D. Hemmerling, J. R. Orozco-Arroyave, A. Skalski, J. Gajda, and E. Nöth, "Automatic detection of Parkinson's disease based on modulated vowels," in *Proc. Interspeech*, Sep. 2016, pp. 1190–1194.
[27] R. Doshi, Y. Chen, L. Jiang, X. Zhang, F. Biadsy, B. Ramabhadran, F. Chu, A. Rosenberg, and P. J. Moreno, "Extending parrotron: An end-to-end, speech conversion and speech recognition model for atypical speech," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 6988–6992.
[28] T. Kourkounakis, A. Hajavi, and A. Etemad, "FluentNet: End-to-end detection of stuttered speech disfluencies with deep learning," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 2986–2999, 2021.
[29] Y. Li, Y. Liu, W.-G. Cui, Y.-Z. Guo, H. Huang, and Z.-Y. Hu, "Epileptic seizure detection in EEG signals using a unified temporal-spectral squeeze-and-excitation network," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 4, pp. 782–794, Apr. 2020.
[30] W. Ko, E. Jeon, S. Jeong, and H.-I. Suk, "Multi-scale neural network for EEG representation learning in BCI," *IEEE Comput. Intell. Mag.*, vol. 16, no. 2, pp. 31–45, May 2021.
[31] P. Janbakhshi, I. Kodrasi, and H. Bourlard, "Automatic dysarthric speech detection exploiting pairwise distance-based convolutional neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Toronto, ON, Canada, Jun. 2021, pp. 7328–7332.
[32] J. Li, R. Zhang, and Y. Li, "Multiscale convolutional neural network for the detection of built-up areas in high-resolution SAR images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 910–913.
[33] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.
[34] R. G. Brereton and G. R. Lloyd, "Support vector machines for classification and regression," *Analyst*, vol. 135, no. 2, pp. 230–267, 2010.
[35] G. Guo, S. Z. Li, and K. Chan, "Face recognition by support vector machines," in *Proc. 4th IEEE Int. Conf. Autom. Face Gesture Recognit.*, Dec. 2000, pp. 196–201.
[36] K. Keng Ang, Z. Yang Chin, H. Zhang, and C. Guan, "Filter bank common spatial pattern (FBCSP) in brain-computer interface," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jun. 2008, pp. 2390–2397.
[37] P. Shen, Z. Changjun, and X. Chen, "Automatic speech emotion recognition using support vector machine," in *Proc. Int. Conf. Electron. Mech. Eng. Inf. Technol.*, Aug. 2011, pp. 621–625.
[38] M. O. Khairandish, M. Sharma, V. Jain, J. M. Chatterjee, and N. Z. Jhanjhi, "A hybrid CNN-SVM threshold segmentation approach for tumor detection and classification of MBI brain images," *Intermediate-Range Ballistic Missile*, vol. 43, no. 4, pp. 743–746, Jun. 2021.
[39] F. Rudzicz, P. van Lieshout, G. Hirst, G. Penn, F. Shein, and T. Wolff, "Towards a comparative database of dysarthric articulation," in *Proc. ISSP*, Strasbourg, France, Dec. 2008, pp. 1–10.
[40] H. Kim, M. Hasegawa-Johnson, A. Perlman, J. Gunderson, T. S. Huang, K. Watkin, and S. Frame, "Dysarthric speech database for universal access research," in *Proc. Interspeech*, Sep. 2008, pp. 1741–1744.
[41] J. Millet and N. Zeghidour, "Learning to detect dysarthria from raw speech," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 5831–5835.
[42] P. Thuwajit, P. Rangpong, P. Sawangjai, P. Autthasan, R. Chaisaen, N. Banluesombatkul, P. Boonchit, N. Tatsaringkansakul, T. Sudhawiyangkul, and T. Wilaiprasitporn, "EEGWaveNet: Multiscale CNN-based spatiotemporal feature extraction for EEG seizure detection," *IEEE Trans. Ind. Informat.*, vol. 18, no. 8, pp. 5547–5557, Aug. 2022.
[43] K. Phapatanaburi, L. Wang, Z. Oo, W. Li, S. Nakagawa, and M. Iwahashi, "Noise robust voice activity detection using joint phase and magnitude based feature enhancement," *J. Ambient Intell. Humanized Comput.*, vol. 8, no. 6, pp. 845–859, Nov. 2017.
[44] L. Wang, K. Phapatanaburi, Z. Go, S. Nakagawa, M. Iwahashi, and J. Dang, "Phase aware deep neural network for noise robust voice activity detection," in *Proc. ICME*, 2017, pp. 1087–1092.
[45] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, May/Jun. 2007, doi: 10.1109/MCSE.2007.55.
[46] L. Van der Maaten and G. Hinton, "Visualizing data using T-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, p. 728, 2008.
[47] S. Tirronen, S. R. Kadiri, and P. Alku, "The effect of the MFCC frame length in automatic voice pathology detection," *J. Voice*, vol. 6, no. 3, pp. 297–440, 2022.
[48] K. Phapatanaburi, W. Pathonsuwan, L. Wang, P. Anchuen, T. Jumphoo, P. Buayai, M. Uthansakul, and P. Uthansakul, "Whispered speech detection using glottal flow-based features," *Symmetry*, vol. 14, no. 4, p. 777, Apr. 2022.

WONGSATHON PATHONSUWAN received the B.E. degree in telecommunication engineering and the M.E. degree in telecommunication and computer engineering from the Suranaree University of Technology, Thailand, in 2019 and 2021, respectively, where he is currently pursuing the Ph.D. degree in telecommunication and computer.

His research interests include wireless communication, artificial intelligent (AI), machine learning (ML), and artificial neural networks (ANN).

KHOMDET PHAPATANABURI received the B.E. degree in electronic and telecommunication engineering and the M.E. degree in electrical engineering from the Rajamangala University of Technology Thanyaburi (RMUTT), Thailand, in 2010 and 2012, respectively, and the Dr.Eng. degree in information science and control engineering from the Nagaoka University of Technology (NUT), Japan, in 2017.

In 2018, he joined the Department of Telecommunication Engineering, Rajamangala University of Technology Isan, as a Lecturer, and became an Assistant Professor, in 2020. His main research interest includes audio and brainwave classification. During his study in Japan, he received the Monbukagakusho (MEXT) Scholarship, from 2014 to 2017. He is a Reviewer for several international journals including IEEE Signal Processing Letters, *Computer Speech and Language*, *APSIPA Transactions on Signal and Information Processing*, and IEEE Access.

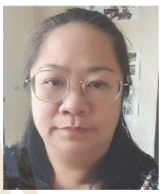
PRAWIT BUAYAI received the B.E. and M.E. degrees in computer engineering from Khon Kaen University (KKU), Thailand, in 2013 and 2016, respectively, and the Ph.D. degree in computer engineering from the University of Yamanashi, Japan, in 2022.

He is currently a Specially Appointed Assistant Professor with the Department of Computer Science and Engineering, University of Yamanashi, and an AI and the IoT Engineer at Mirapro Company Ltd., Japan. His main research interest includes the application of artificial intelligence technology to improve accessibility and productivity in the agriculture and manufacturing.

TALIT JUMPHOO received the B.E. degree in telecommunication and electronic engineering and the Ph.D. degree in telecommunication engineering from the Suranaree University of Technology, Thailand, in 2014 and 2022, respectively. He is currently a Postdoctoral Researcher at the School of Telecommunication Engineering, Institute of Engineering, Suranaree University of Technology. His research interests included biosignal processing, biomedical devices, brainwave classification, and applied machine learning.

PATIKORN ANCHUEN received the B.E. degree in telecommunication engineering from the King Mongkut's Institute of Technology Ladkrabang, Thailand, in 2014, and the M.E. degree in telecommunication engineering and the Ph.D. degree in telecommunication and computer engineering from the Suranaree University of Technology, Thailand, in 2017 and 2020, respectively.

He is currently a Lecturer at the Office of Graduate Studies, Navaminda Kasatriyadhiraj Royal Air Force Academy, Thailand. His research interests include wireless communication, artificial intelligent (AI), machine learning (ML), artificial neural networks (ANN), deep reinforcement learning (DRL), genetic algorithm (GA), particle swarm optimization (PSO), mobile networks, quality of experience (QoE), and 5G communications.



MONTHIPPA UTHANSAKUL (Member, IEEE) received the B.E. degree in telecommunication engineering from the Suranaree University of Technology, Thailand, in 1997, and the M.E. degree in electrical engineering from Chulalongkorn University, Thailand, in 1999, and the Ph.D. degree in information technology and electrical engineering from The University of Queensland, Australia, in 2007.

She is currently an Associate Professor with the Telecommunication School, Suranaree University of Technology. Her research interests include wideband/narrowband smart antennas, automatic switch beam antenna, DOA finder, microwave components, application of smart antenna, and advance wireless communications. She received the Second Prize of Young Scientist Award from 16th International Conference on Microwaves, Radar, and Wireless Communications, Poland, in 2006.

PEERAPONG UTHANSAKUL (Member, IEEE) received the B.E. and M.E. degrees in electrical engineering from Chulalongkorn University, Bangkok, Thailand, in 1996 and 1998, respectively, and the Ph.D. degree in information technology and electrical engineering from The University of Queensland, Brisbane, QLD, Australia, in 2007.

From 1998 to 2001, he was employed as a telecommunication engineer at one of the leading telecommunication companies in Thailand. He is currently working as an Associate Professor and the Dean of the Research Department, Suranaree University of Technology, Nakhon Ratchasima, Thailand. He has more than 100 research publications and the author/coauthor of various books related to MIMO technologies. His research interests include green communications, wave propagation modeling, MIMO, massive MIMO, brain wave engineering, OFDM and advanced wireless communications, wireless sensor networks, embedded systems, the Internet of Things, and network security. He has won various national awards from the Government of Thailand for his contributions and motivation in the field of science and technology. Furthermore, he is the Editor of *Suranaree Journal of Science and Technology* and other leading Thailand journals related to science and technology.

BIOGRAPHY

Mr. Wongsathon Pathonsuwan was born in Nakhon Ratchasima, Thailand, in 1997. He graduated with a bachelor's degree and a master's degree of Engineering in Telecommunication and Computer Engineering in 2019 and 2021, respectively. He is pursuing his Ph.D. in telecommunication and computer engineering at Suranaree University of Technology, Nakhon Ratchasima, Thailand. His research interests include wireless communication, Artificial Neural Network (ANN), Artificial intelligence (AI), Machine Learning (ML), and Deep Learning (DL).