AROA based Pre-Trained Model of Convolutional Neural Network for Voice Pathology Detection and Classification

[1]Manikandan J, [2]Kayalvizhi K, [3]Yuvaraj Nachimuthu and [4]Jeena R
[1,2]Department of Electronics and Communication Engineering, Sri Sairam Engineering College, Chennai, India.
[3]Senior System Operation Engineer, Wells Fargo India Solution Pvt Ltd, Bangalore, India.
[4]Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamil Nadu, India.
[1]manikandan.mangai@gmail.com, [2]kayalvizhikcs@gmail.com, [3]yuvaraj.nachimuthu@gmail.com,
[4]jeenar.sse@saveetha.com

Correspondence should be addressed to Manikandan J : manikandan.mangai@gmail.com.

Article Info
Journal of Machine and Computing (http://anapub.co.ke/journals/jmc/jmc.html)
Doi: https://doi.org/10.53759/7669/jmc202404044
Received 31 August 2023; Revised from 25 October 2023; Accepted 30 January 2024.
Available online 05 April 2024.

Abstract – With the demand for better, more user-friendly HMIs, voice recognition systems have risen in prominence in recent years. The use of computer-assisted vocal pathology categorization tools allows for the accurate detection of voice pathology diseases. By using these methods, vocal disorders may be diagnosed early on and treated accordingly. An effective Deep Learning-based tool for feature extraction-based vocal pathology identification is the goal of this project. This research presents the results of using EfficientNet, a pre-trained Convolutional Neural Network (CNN), on a speech pathology dataset in order to achieve the highest possible classification accuracy. An Artificial Rabbit Optimization Algorithm (AROA)-tuned set of parameters complements the model's mobNet building elements, which include a linear stack of divisible convolution and max-pooling layers activated by Swish. In order to make the suggested approach applicable to a broad variety of voice disorder problems, this study also suggests a unique training method along with several training methodologies. One speech database, the Saarbrücken voice database (SVD), has been used to test the proposed technology. Using up to 96% accuracy, the experimental findings demonstrate that the suggested CNN approach is capable of detecting speech pathologies. The suggested method demonstrates great potential for use in real-world clinical settings, where it may provide accurate classifications in as little as three seconds and expedite automated diagnosis and treatment.

Keywords – Artificial Rabbit Optimization Algorithm, Saarbrücken Voice Database, Convolutional Neural Network, Voice Recognition Systems, Separable Convolution.

I. INTRODUCTION

One of the most fundamental ways that people express themselves is through speech. People with voice abnormalities, however, are unable to fully use this basic human talent [1]. The quality of life for people impacted by voice problems can be improved via early identification and diagnosis, which can lead to effective treatment and management [2]. On top of that, laryngitis, malignancies, and vocal cord paralysis are just a few of the many potential causes of voice pathology, a prevalent illness that impacts many people around the globe. Timely diagnosis is crucial for efficient treatment of voice disorders, which can significantly damage afflicted people' quality of life [3]. The key to successful treatment and management of voice pathology is early and precise diagnosis [4]. Recently, deep learning methods have demonstrated significant potential for enhancing the efficacy and precision of systems that identify speech disorder. Machine learning methods have become more popular for detecting speech pathologies in the past several years [5]. Convolutional neural networks (CNNs) and long short-term networks are two of these methods that have shown outstanding performance in voice processing applications [6].

Since they do not rely on an individual's judgment, these assessment techniques are objective. Plus, they're simple to implement because several online recoding apps make it possible to access speech recordings from anywhere [7]. Hence, in order to reliably differentiate between healthy individuals and those with voice pathologies, several studies have devised vocal processing methods to ascertain which aspects of vocal pathology, when combined with a method, can effectively detect voice pathology automatically within a single framework [8]. For the purpose of conducting objective evaluations of vocal pathology, several databases have been utilized in the literature [9]. Saarbruecken Voice are the most popular databases in the field of voice pathology [10]. Researchers often examine the vocalization of the vowel /a/ as it is accessible in several language databases. Researchers study different vowel pairings as well. One thing to keep in mind is that most researchers in the field of vocal disorders have only included datasets related to certain diseases.

Thanks to developments in signal processing, ML, and DL, vocal pathology detection has been a hot topic as of late [11]. Research on pathology identification has mostly relied on a two-stage pipeline technique. The first step is feature extraction, which involves selecting characteristics to use in compressing acoustic speech waveforms [12]. Using a model based on machine learning or deep learning, a classifier in the second stage predicts the output from the input characteristics. The construction of automatic detection systems requires supervised training of the system utilizing labeled data illustrating normal and abnormal speech patterns [13]. Due to the fact that DL models are notorious for requiring massive amounts of data for accurate training, especially when applied to the classifier stage, massive volumes of speech data are required during the system training stage [14]. There are a to collecting more data, including different recording conditions, speakers' physical limitations, huge variations in disorder characteristics and severity between patients, and privacy issues related to patient data [15]. As a result, current pathological voice databases usually only have a small volume of data. The lack of extensive training data might hinder the creation of reliable and applicable systems for detecting vocal pathology. A linear stack of Swish is the new model architecture that the suggested model employs to address the issues. The AROA model is used to fine-tune the suggested model. **Section 2** lists the relevant works, **Section 3** provides a brief explanation of the model, **Section 4** shows the findings and analysis, and **Section 5** draws the conclusion.

.

II. RELATED WORKS

An algorithm for cognitively healthy and diseased speech classification was developed by Rehman et al., [16]. We used a mix of public and private datasets, including the Saarbruecken voice dataset (SVD), the Massachusetts), and a few sets of voices from different people (healthy and sick) to accomplish this work. To further investigate this matter, we utilized a number of machine learning algorithms—including decision tree, random forest, and support vector machine—to compare and contrast their performance in identifying voice problems. We conduct our experimental studies using the following metrics: recall, sensitivity, accuracy, specificity, F-score, and receiver operating characteristic area. The results showed that algorithm was the most effective in identifying speech disorders, with the accuracy relying on the features chosen using suitable feature selection methods.

An interpretable neural network design, the Interpretable Extraction Network (IMBFN), was suggested by Zhao et al., [17] to enhance the efficacy and generalizability of APVD. This architecture is built on transparent feature extraction logic and a thorough technique for judging results. As a front-end frequency division network, IMBFN proposed and (AT-SincNet) filter bank. Further, in order to extract useful features from speech recordings, IMBFN employed a feature extractor based on a convolutional neural network (CNN) that was developed with two paths and one dimension of depthwise separatability. The synthetic pathology of the voice was determined by analyzing the categorization findings of each voice frame. The MEEI, SVD, and HUPA databases were used to conduct comparative tests. Best improvements in accuracy were 0.1705, F1-score was 0.1977, and Matthews correlation coefficient (MCC) was 0.4463. Accuracy, F1-score, and MCC were measured in 0.7594, 0.8491, and 0.2981, respectively, in blind tests conducted on volunteers from the First University. The results showed that compared to the current approaches, IMBFN offered greater generalization performance, a good APVD effect, and relevant explanations.

Automatic categorization of neurological voice problems might benefit from characteristics based on wavelet scattering transform (WST), as suggested by Yagnavajjula et al., [18]. To provide maintain discriminability across classes while minimizing differences within a class, WST processes speech signals in stages, with each stage containing three operations: convolution, modulus, and averaging. Speech signals of healthy speakers from the Saarbruecken database were used to extract the suggested WST-based characteristics, while speech signals of patients suffering from spasmodic or recurrent laryngeal nerve palsy (RLNP) were also considered. There were three different types of classification tasks: two binary ones (healthy vs. SD and healthy vs. RLNP), and one multi-class one (healthy vs. SD vs. RLNP). The WST-based features were used to train two machine learning algorithms, a feed forward neural). In all three challenges, WST-based features performed better than state-of-the-art features. In addition, the NN classifier that was trained utilizing features based on WST had the greatest overall classification performance.

The use of vocal tract acoustic measures for the diagnosis of voice problems, particularly dysphonia, has been the primary emphasis of Mishra and Sharma [19]. Serious consequences, such as laryngeal cancer, can impact health and quality of life if dysphonia, a communication impairment, is not identified at an early stage. We used the following sources: the Saarbrucken Voice Database for 52 people (26 dysphonic and 26 healthy), the CSL 4500 tool for 28 live samples, and the VOICED Database for 169 subjects (111 dysphonic and 58 healthy). Using these audio recordings, we were able to determine five acoustic parameters: pitch, formants, jitter, and shimmer. With an accuracy of about 85%, the results show that these basic characteristics have promise for the accurate diagnosis of dysphonia and other vocal problems.

To enhance the efficiency of non-invasive systems for detecting speech pathologies, Mohammed et al. [20] presented a new deep Multi-Modal. MMHFNet integrates voice and EGG signals, two complimentary modalities, both at once. In order to make the most of the spatio-spectral information from various levels for multi-layer fusion, it vertically integrates the low-level features that are retrieved from shallow layers with the high-level features that are extracted from deep layers. To diagnose the voice pathology, the characteristics generated by MMHFNet are then inputted into an LSTM classification network. In order to assess how well the planned MMHFNet works, extensive tests are carried out on the open-source Saarbruecken Voice Database (SVD). One way to utilize this dataset is with all of its samples, and another is to create the biggest balanced SVD dataset with selected samples. Results from experiments show that the proposed MMHFNet gets 91% accuracy on all datasets and 96.05% accuracy on balanced samples.

.

Challenges and Limitations of Voice Pathologies Detection (VPD) Systems

Vocal pattern detection (VPD) systems can analyse user-recorded speech to detect and diagnose voice abnormalities. There are a number of obstacles and restrictions to using these systems, despite the fact that they might enhance the efficiency and accuracy of voice disorder diagnosis. Here are a few examples:

Limited availability of high-quality speech data: In order to train their algorithms, voice pathology detection systems use massive databases of high-quality speech samples. But getting your hands on this kind of information can be tough, especially when it comes to uncommon diseases or communities that are underrepresented in current databases.

Variability in speech patterns: Even among people who have the same condition, there can be a great deal of variation in speech patterns. Voice pathology detection systems may find it challenging to reliably diagnose problems using speech samples alone due to this.

Limited diagnostic capabilities: Although methods for detecting voice pathologies can be useful, they might miss some disorders or their root causes when it comes to diagnosing voice problems..

Need for specialized equipment: The specialised hardware and software needed by many voice pathology detection systems—a microphone, for example, or speech analysis software—can be quite pricey and not always easily accessible.

Lack of user acceptance: Concerns regarding privacy or a preference for human healthcare providers may make some people reluctant to employ vocal pathology detection technologies.

While there is hope for voice pathology detection technologies to streamline and enhance the diagnosis of voice disorders, these tools are not perfect. In order to improve these systems and make them available to people who could use them, researchers and developers need to keep working to solve these problems.

III. PROPOSED MODEL

Database

German researchers at Saarland University's Phonetics Research Institute created the SVD, which includes the phrase "Guten Morgen, wie geht es Ihnen?" and over 2000 voiced samples of $sustained\ /a/,/i/, and\ /u/$ vowels [21]. The recordings have a resolution of 16 bits and are sampled at 50 kHz. Speech files that contain sustained vowels typically range in duration from one second to three seconds. The dataset, on the other hand, includes recordings of the/a/, /i/, and /u/ vowels made by 687 healthy voices and 1354 voices affected by one of seventy-one distinct diseases. The dataset information for the SVD used in this investigation is displayed in **Table 1**. Despite an imbalance in the numbers between the normal and diseased voice samples, this research proposes oversampling to bring them into balance. Also, the 687 healthy samples are included in the balanced set, and the diseased samples are identical in both datasets.

Table 1. Sum of Examples in the Experimental Dataset.

	Balanced Class	Imbalanced Class
Number of normal voices	1354	687
Number of pathological voices	1354	1354

Data Preprocessing

The pre-processing procedure is crucial for cleaning up the input data and improving the classification accuracy..

Noise Injection

Depending on maintained, the noise can random using the white noise function in the NumPy package. This method simply inserts a random value into the data. When noise is injected into a neural network model during its training phase, it produces a regularisation effect that increases the model's robustness.

Time Shifting

The audio is shifted to the left or right at a random interval of one second. Moving the playhead to the left (fast forward) will label the first x). The last x seconds are indicated as 0 (i.e., silence) if the right (backwards and ahead).

Changing Pitch

This method, which is commonly used in musical instruments, has a real-world application. It is a way to change the pitch of a sound without changing its velocity. It is implemented via Librosa's pitch function. It raises or lowers the angle by a given, arbitrary value. The procedure can't be put into action without wave samples and sampling speeds..

Stretch (Speed)

Time stretching refers to the process of changing the duration or tempo of an audio broadcast without changing its pitch. On the other hand, pitch scaling is when the speed of a sound is changed while keeping the pitch constant. One pitch scaling effect that effects units use is pitch shift, which is ideal for use during live performances. The pitch control procedure allows the user to simultaneously change the recording's speed and pitch, allowing for a more natural and organic listening experience.

Feature Extraction

Here we provide a quick overview of the theory behind the MFCC and LPC feature parameters. The MFCC method has become the de facto standard for automated VPD feature extraction. Preprocessing, fast Fourier are the steps that make up the process. The first step is to frame and window the voice signal before highlighting it. After that, the magnitude spectrum is retrieved using a Fourier analysis that just takes a few seconds. Using 24 overlapping triangular spectrum is converted into a mel spectrum with an equal centre frequency distribution. It is computed the square of the mel spectrum, which is the logarithm of the output of each filter bank. Lastly, the DCT is used to extract the 20th-order MFCCs via the log power.

Through the use of linear predictive analysis, the vocal tract information of a particular speech can be effectively extracted. In other words, LPCs stand for consistent and periodic source behaviours. The commonly employed in speech signal processing is linear prediction (LP). The idea behind LP is that it is possible to approximate a voice sample by linearly combining samples from earlier iterations. Then, for a finite period, we minimise the sum of the squared discrepancies between the predicted samples to establish a unique set of prediction coefficients. Obtaining a glottal excitation and vocal tract parameters are two separate parts of LP analysis. The LP coefficients are one kind of variable and the LP residuals are another. Presumably, impulse trains and random noise are used to generate voiced and unvoiced speech, respectively. For each speech frame, the 20th MFCC and LPC parameters are calculated during the feature extraction step in this research (window size = 40 ms;

Construction of Fully Connected Output Layers

Our suggested model's first output was enhanced with fully linked layers. You will find the following components in fully linked layers. The research began by applying the flatten technique to the output in Pytorch. Next, dropout was employed. Then, to make the result more concise, a linear layer was used. Lastly, the activation function ReLu was employed. Small positive or negative weights can speed up training with the Rectified Linear Activation Function, or ReLu. Furthermore, it reduces the output's dynamic range. Applying the 1D batch normalisation was done. Because the input layers are normalised for each mini-batch by rescaling and refactoring, the deep CNN becomes quicker and more reliable. On top of that, it shortens the training period of the model.

The Xception model's robustness was a cornerstone of our suggested model design. We found that Xception trains quickly and is a rapid model overall. Our exploration of the Efficient Nett architecture led us to the conclusion that the MB convolution block serves as the model's foundational component. An initial 2D separable convolution layer using the ReLu activation function is the first component of the original MB convolution block. Subsequent layers include a 2D depth-wise convolution layer also using the ReLu activation function, a global averaging pooling layer, and finally, a 2D convolution layer for pressing and expanding. The study led to the development of the 48-layer DFN. The components of these strata number seventeen. The design consists of an entrance, a central area, and an exit. The incoming flow would not be complete without the max-pooling layer..

Following two conventional convolution layers comes the fundamental entrance flow block, which is repeated three times. The kernel performs a hop of two rows and columns during convolution since each of them has a stride of two. The central flow features eight repetitions of the fundamental building component. If you repeat a block, its output will be considered input the following time around. We improved our architecture's accuracy and feature extraction capabilities by including three MB convolution blocks in this intermediate phase. Based on the results of the studies, we also enhanced the middle flow of our design by adding three MB Convolutions. Our goal is to build a generalised model architecture that is both durable and efficient, all while keeping the model's computational complexity low.

The exit flow begins with the fundamental construction block and continues with a fully linked layer and a global pooling layer. After each of these layers has finished processing the image, the classifier takes in the data and assigns a label to it. In order to categorise the image, we employed XGBoost as the classifier. Firstly, our proposed model uses Image-Net weights as initial weights instead of random weights, which enhances performance and speeds up convergence. For computational efficiency, our model employs depth-wise convolution rather than conventional convolutions. The conventional convolution layer is more computationally costly and time-consuming due to the higher number of multiplication operations. All input channels undergo filtering in typical convolution, and their values are combined in a single step.

This process, on the other hand, requires two phases when using depth-wise separable convolution. In the filtering stage, it does depth-wise convolution, the first step. In contrast to the usual method of applying convolution to all input channels at once, it only does it to one. All that is involved in the convolution procedure is adding and multiplying elements one by one. Step two involves the combining procedure, which is carried out using point-wise convolution. In point-wise convolution, the outputs of the layers are linearly combined. Computationally efficient and significantly less expensive (in terms of multiplications) is depth-wise separable convolution. All of the layers process the input picture using depth-wise separable convolution. In order to train the model more quickly and with fewer dimensions, max pooling and global average pooling layers are used. To advance the model's computing efficiency and decrease the number of multiplication operations, the two-step technique was employed.

There are three parts to an MB Convolution block. You can see residual connections in each of these blocks all the way through. Assuming there are several layers following the initial one, each of which needs to improve upon the traits discovered by the first, as well as discover new ones and store them. For the classifier to make use of characteristics from previous levels, the remaining connections between the first and second layers are crucial. A 1 x 1 convolution is performed on the input in the MB Convolution block. The purpose of using this convolution is to increase the dimensionality of the input. Afterwards, depth-wise convolution is used on that dataset. Afterwards, the result of depth-wise convolution is subjected to 1 × 1 convolution once again. This time, it's for reversing the data's transformation into its original, input-specific dimensions. After that, we total up all of these outputs. "Expand and squeeze" is another name for this entire procedure.

There is said to be a two-stage procedure to creating features. The first phase is feature aggregation, which involves grouping features that are comparable. The second step involves processing each group independently to produce new features. At the outset, feature aggregation is thought of as 1 × 1 convolution, whereas feature formation is thought of as depth-wise convolution. Since the 1 × 1 convolution eliminates several multiplication processes, it is both quicker and less expensive. To get around the complexity that comes with a deeper model, depth-wise convolution is employed. In order to improve the feature set, each MB convolution block projects its output to the high-dimensional block, which is responsible for creating and aggregating features. Therefore, our model maintains its computational efficiency and speed even after including the layers. In addition, we conducted experiments and confirmed that the Swish activation function outperforms the ReLu activation function; switching from ReLu to Swish resulted in a 0.7 improvement in performance. Consequently, our suggested architecture made use of the Swish activation function following all of the levels. Next comes batch normalisation for every convolution, separable, and depth-wise convolution layer [22]. A depth-wise convolution layer was employed because it takes depth into account along with other spatial dimensions such as width and height. There are three channels in an input picture that represent the pixel's red, green, and blue values. The number of channels, however, grows after a certain number of convolutions. One way to look at each channel is as a picture interpretation.

We tried several various configurations using MB convolution blocks. To keep the model's complexity to a minimal, though, we utilised three MB convolution blocks. To facilitate feature extraction, these blocks are incorporated into our architecture's middle layer. To avoid depth expansion, all convolution layers employ a depth multiplier of one. While there are 25 million parameters, it's significantly less than the Efficient Nett model but somewhat more than the Xception model.

Training and Validation Stage

As said before, the dataset has been divided into several sets for the purposes of training and validating the deep learning model. Because we want to utilise the same data sets for all of our experiments, we trained and validated each set with 10-fold cross-validation indices. We saved the test set for last-ditch model testing. Following this, the testing and validation sets are divided into age and gender categories according to medical status. In order to avoid any leaks into set, the lengthy recordings were segmented into many pieces. These pieces have been meticulously taken out of the set. In the training set, the additional chunks were also added. We employed a set number of samples from 150 healthy (H) and pathological (P) samples at each position in the validation confusion matrix. We tested the confusion matrix with 874 diseased samples and 200 healthy samples to see if the suggested model could detect a pathology. To ensure that there was an equal number of healthy and diseased samples in each set, we partitioned the dataset into validation, and testing. We have included the leftovers in our test set. The total number of samples utilised for training, validation, and testing is 960 (480 healthy and 480 pathologic), 300 (150 healthy and 150 pathogenic), and 874 (200 healthy and 674 pathologic). The samples are not evenly distributed during the training phase. Our solution was to make adjustments to the sample weights that are utilised during training for the minority groups in order to make up for this. The ultimate sample weight is a three-part weight product. Increasing partial weight quantifies the number of subgroups in the selected group, such as the ratio of normal to pathological. We achieved this by introducing a class weight α, a gender weight β, and a set of gender-age weight γ, which together formed a final sample weight!, which is computed as! = α·β·γ. Also, for every given sample, weights may be found in subgroups αi of group α, βi of group β, and γi of group γ. The optimal hyperparameters for the cross-validation setup have been selected for use as a performance metric. We have adjusting them, and then we tested them throughout the whole training set. The end findings are given in the form of a confusion matrix (CM) and a classification report (CR). The F1 score (which takes into account both average accuracy and recall) and precision are determined by the CR tables according to formulas 1, 2, and 3.

Fine-tuning using Artificial Rabbits Optimization (ARO)

The ARO algorithm is primarily based on two natural principles of rabbit survival: random hiding and detour foraging [23]. One of their exploring strategies, detour foraging, involves the rabbits eating grass close to the nest in order to evade natural predators. When rabbits engage in random hiding, they typically wander to different burrows in order to increase their level of concealment. The initialisation procedure is the backbone of every search algorithm. Assuming that d is the dimension of the design variable, N is the size of the artificial rabbit colony, and ub and lb are the bounds, respectively. The following is the procedure for initialisation:

$$\vec{z}_{i,k} = r.(ub_k - lb_k) + lb_k, k = 1,2,\ldots,d \tag{1}$$

where $\vec{z}_{i,k}$where r is a supplied random integer and signifies the location of the jth dimension of the ith rabbit. Detour foraging focuses on the exploration phase, whereas the metaheuristic algorithm primarily takes into account the exploitation phase. In detour foraging, each rabbit will often investigate a different spot in the group at random in order to find food, rather than sticking to the original plan. What follows is the revised formula for detour foraging.

$$\vec{v}_i(t+1) = \vec{z}_j(t) + R.\left(\vec{z}_i(t) - \vec{z}_j(t)\right) + round\left(0.5.(0.05 + r_1)\right).n_1 \tag{2}$$

$$R = l.C \tag{3}$$

$$l = \left(e - e^{\left(\frac{t-1}{T_{max}}\right)^2}.\sin(2\pi r_2)\right) \tag{4}$$

$$C(k) = \begin{cases} 1 & if \;\; k == G(l) \\ 0 & else \end{cases} \; lk = 1,\ldots,d \; and \; l = 1,\ldots,[r_3.d] \tag{5}$$

$$G = randp(d) \tag{6}$$

$$n_1 \sim N(0,1) \tag{7}$$

where $\vec{v}_{i,k}(t+1)$ denotes the novel site of rabbit, $i,j = 1,\ldots,N$. $\vec{z}_i$ denotes the site of the ith artificial rabbit, and $\vec{z}_j$ characterizes artificial rabbits at other random sites. $T_{\max}$ is the maximum sum of iterations. [·] symbolizes function, which characterizes rounding to the nearest integer, and $randp$ embodies a stochastic arrangement from 1 to d integers. r_1, r_2, and r_3 are stochastic numbers from 0 to 1. L signifies the running length, It represents the rate of movement during detour foraging. The standard normal distribution is followed by n1. The random integer n_1, which follows a normal distribution, mostly represents the disturbance. To aid ARO in avoiding the local extremum and conducting a global search, the final term of Equation (2) can be perturbed.

Rabbits often dig many burrows surrounding their nests and pick one at random to hide in, reducing the likelihood of being preyed upon. This is mostly based after the exploration step of the algorithm. We begin by outlining how rabbits come up with burrows at random. The rabbit with the ith leg creates the jth burrow by

$$\vec{b}_{i,j}(t) = \vec{z}_i(t) + H.g.\vec{z}_i(t) \tag{8}$$

$$H = \frac{T_{max}-t+1}{T_{max}}.n_2 \tag{9}$$

$$n_2 \sim N(0,1) \tag{10}$$

$$g(k) = \begin{cases} 1 & if \;\; k == k \\ 0 & else \end{cases} \; l\,k = 1,..,d \tag{11}$$

where $i = 1,\ldots,N \, and \, j = 1,\ldots,d,$ and the second variable is normally distributed. As a function of stochastic perturbations, the linearly from 1 to 1/Tmax. The value of a changes throughout 1000 rounds, as seen in **Fig 1**. Throughout the iterations, the chart shows that the H value trend declines, which maintains a balanced exploration to exploitation. Presented below is the formula for the random hiding method's update.

.

$$\vec{v}_i(t+1) = \vec{z}_i(t) + R.(r_4.\vec{b}_{i,r}(t) - \vec{z}_i(t)) \tag{12}$$

$$g_r(k) = \begin{cases} 1 & if \; k == [r_5.d] \\ 0 & else \end{cases} \; l\,k = 1,\ldots,d \tag{13}$$

$$\vec{b}_{i,r}(t) = \vec{z}_i(t) + H.g_r.\vec{z}_i(t) \tag{14}$$

where $\vec{v}_i(t+1)$ is the new site of rabbit,!b i,r(t) characterizes a arbitrarily designated burrow among the d burrows made by hiding, and r_4 and r_5 characterize the random sum assumed by us in the interval 0 to 1. R is assumed by Equations (3)–(6).

We use Equation (15) to reposition the ith artificial rabbit after we've used the two update procedures.$\vec{z}_i(t+1) =$

$$\begin{cases} \vec{z}_i(t) & if \;\; f(\vec{z}_i(t)) \leq f(\vec{v}_i(t+1)) \\ \vec{v}_i(t+1) & else \, f\left(\vec{z}_i(t)\right) > f(\vec{v}_i(t+1)) \end{cases} \tag{15}$$

It is more common for populations to do the exploration phase early in an optimisation algorithm and the exploitation phase midway through and at the end. In order to model the change from exploration to exploitation, ARO models its discovery scheme after the rabbits' energy levels, which naturally decline with time. The energy factor in our method for artificial rabbits is defined as:

$$A(t) = 4.\left(1 - \frac{t}{T_{max}}\right).In\frac{1}{r} \tag{16}$$

using a specified random integer r and a random number between zero and one as parameters. The value of a changes after 1000 rounds, as seen in **Fig 2**. As can be seen from the data in the picture, the overall trend in the A's is decreasing, which helps to keep the transition from exploration to exploitation balanced over the rounds.

IV. EXPERIMENTAL PARAMETERS

We were unable to train our model on any scheme due to the massive size of the dataset. We used a system with a $16-core\ CPU, 128\ GB\ of\ RAM, 200\ GB\ of\ SSD$ persistent disc, and Ubuntu 18.04 LTS for the operating system to build, train, and test our model. We also experimented with different parameters to find the best mixtures of the two models. Our model couldn't have worked quickly or correctly without these settings. When training, we made use of the instance of the GCP AI notebook. Our proposed model's parameters are displayed in **Table 2**. We found that these parameters gave our model the greatest results after testing it with a range of alternative settings.

Table 2. Parameters used in the Projected Model.

Used Value	Parameters
0.450.00130	Drop Out Learning Rate Epochs
128	Batch Size
ARO	Optimizer
Swish	Activation Function

Validation analysis of Proposed Model

Table 3 provides the experimental analysis of planned model with existing procedures in terms of different metrics.

Table 3: Comparative study of Projected Model with Existing Procedures

Methodology	Parameter Evaluation			
	Accuracy	Precision	Recall	F-measure
DarkNet	88.89	79.12	80.92	85.27
LeNet	72.32	80.53	83.69	86.07
ResNet	81.43	82.07	90.06	89.28
VGGNet	87.16	81.04	84.17	83.08
MobileNet	94.38	95.43	96.46	96.34
ARO-EfficientNet	**96.90**	**97.84**	**98.20**	**98.67**

In above **Table 3** characterise that the Comparative study of Predictable prototypical with existing procedures. In the investigation study of DarkNet technique reached the accuracy as 88.89 and precision as 79.12 and recall as 80.92 and then F-measures as 85.27 respectively. Then the LeNet technique got the accuracy as 72.32 and precision as 80.53 and recall as 83.69 and then F-measures as 86.07 respectively. Then the ResNet technique got the accuracy as 81.43 and precision as 82.07 and recall as 90.06 and then F-measures as 89.28 respectively. Then the VGGNet technique reached the accuracy as 81.04 and recall as 84.17 and then F-measures as 83.08 respectively. Then the MobileNet technique reached the accuracy as 94.38 and precision as 95.43 96.46 and then F-measures as 96.34 respectively. Then the ARO-EfficientNet technique stretched the accuracy as 96.90 and precision as 97.84 and recall as 98.20 and then F-measures as 98.67 respectively.

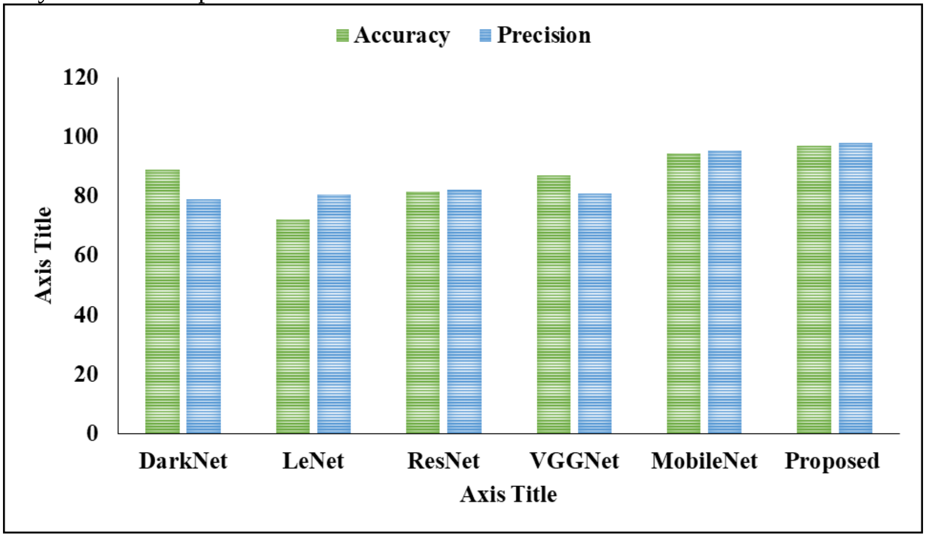

Fig 1. Graphical Analysis of Various Models

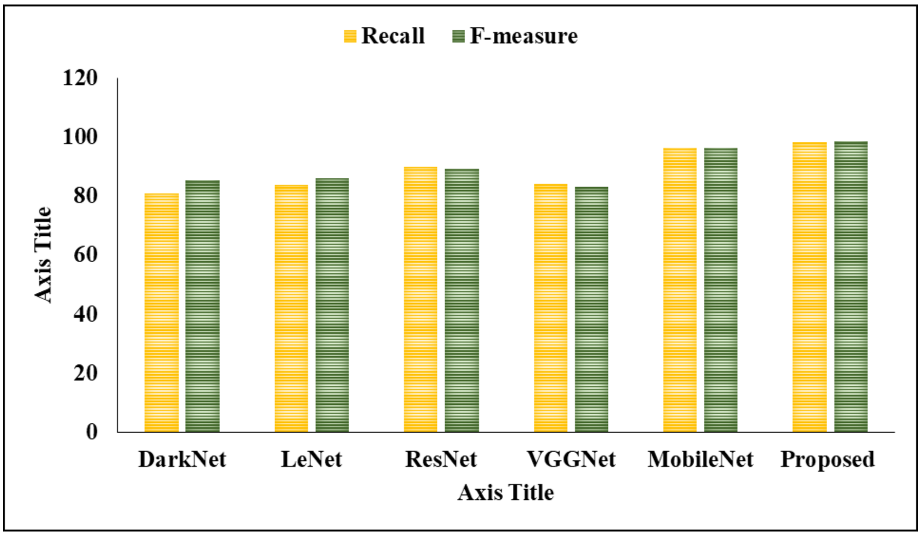

Fig 2. Visual Representation of Different Pre-Trained Models

V. CONCLUSION

It is possible to construct detection systems based on ML and DL that can automatically differentiate between normal and abnormal sounds. In this study, we look for strong answers by exploring ways to make vocal pathology detection more accurate. The lack of accessible, high-quality testing samples is the primary issue preventing researchers in this area from making forward. Traditional dysphonic voice characteristics have been the focus of most relevant research because they are both predictable and clinically interpretable. Due to its difficult and complicated dysphonic speech pattern analysis features, the SVD dataset was chosen for this study. Next, this research presents a new real-time method for detecting speech pathologies using a CNN model, with the ARO model selecting the model's fine-tuning optimum. A Voice Pathology Detection scheme has used the model. The Voice Pathology Detection system's development technique includes steps such as inference. We begin by establishing the relative prediction accuracy of the suggested model by applying the SVD dataset to a pre-trained CNN model. In order to determine if deep learning CNNs are useful for detecting speech pathologies, this research set out to conduct a preliminary investigation. A novel vowel combination and gender separation are two areas that might be considered for improved dataset dimensionality and feature extraction in future study. Also, trying out various CNN and training models can help refine the method for detecting speech pathologies. A poor intelligibility voice severely impairs the social communication skills of people who have had laryngeal cancer; hence, future research may involve applying this strategy to oesophageal sounds. Laryngectomy patients will greatly benefit from any new information on oesophageal speech. In conclusion, the suggested approach shows promise in a real-world clinical setting, where it may provide accurate classifications in as little as three seconds and facilitate rapid, automated diagnosis and treatment.

Data Availability

No data was used to support this study.

Conflicts of Interests

The author(s) declare(s) that they have no conflicts of interest.

Funding

No funding agency is associated with this research.

Competing Interests

There are no competing interests.

References

[1]. L. Geng, Y. Liang, H. Shan, Z. Xiao, W. Wang, and M. Wei, “Pathological Voice Detection and Classification Based on Multimodal Transmission Network,” Journal of Voice, Dec. 2022, doi: 10.1016/j.jvoice.2022.11.018.

[2]. N. Q. Abdulmajeed, B. Al-Khateeb, and M. A. Mohammed, “A review on voice pathology: Taxonomy, diagnosis, medical procedures and detection techniques, open challenges, limitations, and recommendations for future directions,” Journal of Intelligent Systems, vol. 31, no. 1, pp. 855–875, Jan. 2022, doi: 10.1515/jisys-2022-0058.

[3]. L. Chen and J. Chen, “Deep Neural Network for Automatic Classification of Pathological Voice Signals,” Journal of Voice, vol. 36, no. 2, pp. 288.e15-288.e24, Mar. 2022, doi: 10.1016/j.jvoice.2020.05.029.

[4]. R. Islam, E. Abdel-Raheem, and M. Tarique, “Voice pathology detection using convolutional neural networks with electroglottographic (EGG) and speech signals,” Computer Methods and Programs in Biomedicine Update, vol. 2, p. 100074, 2022, doi: 10.1016/j.cmpbup.2022.100074.

[5]. Thirumalraj, V. Asha, and B. P. Kavin, “An Improved Hunter-Prey Optimizer-Based DenseNet Model for Classification of Hyper-Spectral Images,” Advances in Medical Technologies and Clinical Practice, pp. 76–96, Oct. 2023, doi: 10.4018/979-8-3693-0876-9.ch005.

[6]. Ksibi, N. A. Hakami, N. Alturki, M. M. Asiri, M. Zakariah, and M. Ayadi, “Voice Pathology Detection Using a Two-Level Classifier Based on Combined CNN–RNN Architecture,” Sustainability, vol. 15, no. 4, p. 3204, Feb. 2023, doi: 10.3390/su15043204.

[7]. N. Omeroglu, H. M. A. Mohammed, and E. A. Oral, “Multi-modal voice pathology detection architecture based on deep and handcrafted feature fusion,” Engineering Science and Technology, an International Journal, vol. 36, p. 101148, Dec. 2022, doi: 10.1016/j.jestch.2022.101148.

[8]. M. Zakariah, R. B, Y. Ajmi Alotaibi, Y. Guo, K. Tran-Trung, and M. M. Elahi, “An Analytical Study of Speech Pathology Detection Based on MFCC and Deep Neural Networks,” Computational and Mathematical Methods in Medicine, vol. 2022, pp. 1–15, Apr. 2022, doi: 10.1155/2022/7814952.

[9]. Zhou, Y. Wu, Z. Fan, X. Zhang, D. Wu, and Z. Tao, “Gammatone spectral latitude features extraction for pathological voice detection and classification,” Applied Acoustics, vol. 185, p. 108417, Jan. 2022, doi: 10.1016/j.apacoust.2021.108417.

[10]. S. Tirronen, S. R. Kadiri, and P. Alku, “The Effect of the MFCC Frame Length in Automatic Voice Pathology Detection,” Journal of Voice, Apr. 2022, doi: 10.1016/j.jvoice.2022.03.021.

[11]. F. Javanmardi, S. R. Kadiri, M. Kodali, and P. Alku, “Comparing 1-dimensional and 2-dimensional spectral feature representations in voice pathology detection using machine learning and deep learning classifiers,” Interspeech 2022, Sep. 2022, doi: 10.21437/interspeech.2022-10420.

[12]. S. Fujimura et al., “Classification of Voice Disorders Using a One-Dimensional Convolutional Neural Network,” Journal of Voice, vol. 36, no. 1, pp. 15–20, Jan. 2022, doi: 10.1016/j.jvoice.2020.02.009.

[13]. F. Javanmardi, S. R. Kadiri, and P. Alku, “A comparison of data augmentation methods in voice pathology detection,” Computer Speech &amp; Language, vol. 83, p. 101552, Jan. 2024, doi: 10.1016/j.csl.2023.101552.

[14]. N. Q. Abdulmajeed, B. Al-Khateeb, and M. A. Mohammed, “Voice pathology identification system using a deep learning approach based on unique feature selection sets,” Expert Systems, May 2023, doi: 10.1111/exsy.13327.

[15]. Fu, X. Zhang, D. Chen, and W. Hu, “Pathological Voice Detection Based on Phase Reconstitution and Convolutional Neural Network,” Journal of Voice, Oct. 2022, doi: 10.1016/j.jvoice.2022.08.028.

[16]. M. Ur Rehman, A. Shafique, Q.-U.-A. Azhar, S. S. Jamal, Y. Gheraibia, and A. B. Usman, “Voice disorder detection using machine learning algorithms: An application in speech and language pathology,” Engineering Applications of Artificial Intelligence, vol. 133, p. 108047, Jul. 2024, doi: 10.1016/j.engappai.2024.108047.

[17]. Zhao, Z. Qiu, Y. Jiang, X. Zhu, X. Zhang, and Z. Tao, “A depthwise separable CNN-based interpretable feature extraction network for automatic pathological voice detection,” Biomedical Signal Processing and Control, vol. 88, p. 105624, Feb. 2024, doi: 10.1016/j.bspc.2023.105624.

[18]. M. K. Yagnavajjula, K. R. Mittapalle, P. Alku, S. R. K., and P. Mitra, “Automatic classification of neurological voice disorders using wavelet scattering features,” Speech Communication, vol. 157, p. 103040, Feb. 2024, doi: 10.1016/j.specom.2024.103040.

[19]. J. Mishra and R. K. Sharma, “Vocal Tract Acoustic Measurements for Detection of Pathological Voice Disorders,” Journal of Circuits, Systems and Computers, Jan. 2024, doi: 10.1142/s0218126624501731.

[20]. H. M. A. Mohammed, A. N. Omeroglu, and E. A. Oral, “MMHFNet: Multi-modal and multi-layer hybrid fusion network for voice pathology detection,” Expert Systems with Applications, vol. 223, p. 119790, Aug. 2023, doi: 10.1016/j.eswa.2023.119790.

[21]. Saveleva., “Graph-based Argument Quality Assessment,” Proceedings of the Conference Recent Advances in Natural Language Processing - Deep Learning for Natural Language Processing Methods and Applications, 2021, doi: 10.26615/978-954-452-072-4_143.

[22]. M. A. Thirumalraj, B. Rajalakshmi, B. S. Kumar, and S. Stephe, “Automated Fruit Identification using Modified AlexNet Feature Extraction based FSSATM Classifier,” Mar. 2024, doi: 10.21203/rs.3.rs-4074664/v1.

[23]. Riad, A. J., Hasanien, H. M., Turky, R. A., & Yakout, A. H. (2023). Identifying the PEM fuel cell parameters using artificial rabbits optimization algorithm. Sustainability, 15(5), 4625.