Contents lists available at GrowingScience

International Journal of Data and Network Science

homepage: www.GrowingScience.com/ijds

Deep convolutional neural network architectures for stuttering classification using spectrograms and audio features

Ibrahim I. M. Manhrawy[a*], Katrina Sundus[b], Hanaa Fathi[c], Arar Al Tawil[a], Amro M. Soliman[d], Abdelmoty M. Ahmed[e,f] and Ahmed Elashry[g,h]

[a]*Faculty of Information Technology, Applied Science Private University, Amman, 11931, Jordan*
[b]*Department of Computer Science, Faculty of Science and Information Technology, Al-Zaytoonah University of Jordan, Amman, Jordan*
[c]*College of Computer Science and Informatics, Amman Arab University, Amman, Jordan*
[d]*Special Education Department, College of Education, King Khalid University, Abha, Saudi Arabia*
[e]*Computer Science Department, Faculty of Information and Technology, Ajloun National University, Ajloun, 26810, Jordan*
[f]*Artificial Intelligence And Digital Transformation Center (AIDTC), Ajloun National University, Ajloun, Jordan*
[g]*Faculty of Computer Studies, Arab Open University, Saudi Arabia*
[h]*Information Systems Department, computers and information Faculty, Kafr El-Sheikh University, Egypt*

C H R O N I C L E

Received August 28, 2025
Received in revised format September 28, 2025
Accepted November 9 2025
Available online November 9 2025

Keywords:
Stuttering
MFCC
CNN
STFT
Spectrogram

A B S T R A C T

Speaking disorders such as stuttering or stammering cause sounds, syllables, or phrases to be repeated or delayed, impairing natural speaking patterns. A person's quality of life may suffer as a result, and communication with others may become challenging. Over 70 million individuals globally experience stuttering. Many people with this illness struggle to fit in and lose confidence while speaking in front of others. Because of this, this research aims to create a thoroughly annotated dataset and, in the end, provide an extensive empirical comparison of deep convolutional neural network (CNN) architectures on the classification of stuttering or Stammering. Short-time fast Fourier transformation (STFT) is used in this study to convert audio data into a visual representation (spectrogram), and Mel Frequency Cepstral Coefficients (MFCC) are then used to extract many audio features to use CNNs to create a workable classification system. Matthew's correlation coefficient (MCC), time to build, and accuracy score are used to evaluate the classifiers' performance. The final findings showed that AlexNet, Le-Net5, ResNet50, LSTM-CNN, and VGG are the top five CNN classifiers that have been investigated. The four designs that are being used have an exceptionally high degree of accuracy, ranging from 90% to 97%.

© 2026 by the authors; licensee Growing Science, Canada.

1. Introduction

Speech disorders profoundly affect interpersonal relationships, emotional health, and communication. Stuttering, a widespread speech disorder, manifests as blockages, prolongations, and involuntary sound repetitions. It has a global impact, affecting millions of individuals. The multifaceted nature of stuttering, in conjunction with its diverse presentations, poses difficulties in diagnosis and treatment, underscoring the imperative for inventive strategies in its control (Kumar, Rajpal, & Rathore, 2018). Recent developments in machine learning, specifically Convolutional Neural Networks (CNNs), have demonstrated outstanding potential in speech processing. CNNs, renowned for their capacity to analyze visual stimuli, have been successfully modified to analyze audio, thereby presenting novel approaches to diagnosing and comprehending speech disorders. Their efficacy in deriving significant patterns from intricate data renders them well-suited for classifying interrupted speech instead of everyday speech. This study examines the effectiveness of four convolutional neural network (CNN) architectures (Saadi et al., 2025; Ahmed et al., 2024; Zena et al., 2024) - AlexNet, LeNet, ResNet-50, and VGG-16—in detecting halting speech. It emphasizes the potential of each model in this groundbreaking application (Li, Chan, & Chun, 2010).

* Corresponding author
E-mail address: A. Altawil (ar_altawil@asu.edu.jo)

ISSN 2561-8156 (Online) - ISSN 2561-8148 (Print)
© 2026 by the authors; licensee Growing Science, Canada.
doi: 10.5267/j.ijdns.2025.11.003

The transformative potential of CNNs to enhance diagnostic accuracy and intervention strategies for stuttering serves as our impetus. Through the automation of speech pattern classification, our objective is to enhance the accuracy of stuttering diagnosis, thereby enabling the development of more focused and effective treatment plans. Speech therapists and clinicians could derive substantial benefits from this method (Bagci & Erzin, 2007) in the form of a reliable instrument for tracking patient development and customizing interventions.

There are two primary contributions to this investigation. To begin, a comparative analysis is undertaken to assess the performance of four prominent CNN models in the classification of interrupted speech. This evaluation uses accuracy, F1-Score, precision, and recall metrics to identify the CNN architecture most suited to the task at hand. Furthermore, we present an innovative approach to identify interrupted speech utilizing CNNs; this development signifies a substantial progression in speech pathology. The results of our study not only demonstrate the efficacy of CNNs in the classification of speech disorders and establish a foundation for further investigation into the utilization of deep learning methods in speech therapy (McKay & Fujinaga, 2005).

2. Related works

The research described by Chai and Vercoe (2001) was conducted to differentiate three distinct types of folk music: Irish, German, and Austrian. Several Hidden Markov Model (HMM) architectures were used to assess and compare the dataset in order to find statistical differences between the different people. The categorization performance is 77% on average. By contrast, when compared to HMM, the study conducted by Xu et al. (2005) employing Support Vector Machines as a statistical machine learning approach produced better results in MGC. The utilization of a genetic algorithm to select frequency-domain and low-level features was investigated in a recent study (Girsang et al., 2019). Various classification algorithms are subjected to comparative analysis, including K-nearest neighbor (KNN), SVM, and Naïve Bayes (NB). This investigation utilized samples from the GTZAN dataset (Chai & Vercoe, 2001; Tzanetakis & Cook, 2000) to distinguish among ten genres. SVM achieved an optimal classification accuracy of 80.1% compared to KNN and NB. Similar to how SVM served as the basis for classification in de Sousa et al. (2016), the GTZAN dataset and a recommended Brazilian Music Dataset (BMD) were utilized to evaluate the classification. Six descriptors were used by the authors: high-level, rhythmic, spectral, tonal, time-domain, and sound effect. Moreover, dissonance, pace, key, tonality, sharpness, inharmoniousness, and rhythm histograms are common traits. Compared to GTZAN, BMD yielded a higher accuracy of 86.11%.

Various feature extraction approaches are applied to enable autonomous MGC with a large number of classifiers. These classifiers are the semi-supervised technique (Song & Zhang, 2008), Ada-Boost (Bergstra et al., 2006), radial basis function (RBF) (Li & Ogihara, 2006), and Gaussian mixture model (GMM) (Mouret, 2016). Nonetheless, a thorough analysis of the results of several machine learning classification techniques was also released by Scaringella et al. (2006), suggesting that the search for the ideal classifier is still necessary. Deep learning techniques used in the MGC have had a significant impact on the classification results in recent years. According to Choi et al. (2016), there are a lot of benefits to using deep learning models in the context of MGC. Its use of learnt characteristics from different hierarchical layers of the neural networks rather than derived features taken from structured input is one of its main advantages. The second is the application of SGD models' (Akhtar et al., 2024) hierarchical topology properties to musical analysis across all time and frequency ranges. Numerous researchers have utilized Convolution Neural Networks (CNNs) for MGC (Allamy & Koerich, 2021; Al-Anazi et al., 2024; Cheng et al., 2020; Choi et al., 2016; Sènac et al., 2017). To classify the 1000 audio segments comprising the GTZAN dataset, Allamy and Koerich (2021) proposed a 1D CNN architecture comprising two Convolution layers and nine residual blocks based on the sample level. The proposed 1D CNN demonstrated an accuracy of 80.93%. However, the authors suggest that for future research, larger music datasets, including the LMD dataset (Jr et al., 2008), MSD dataset (Kim et al., 2018), and free music archive dataset (Defferrard et al., 2017), could yield more accurate results. Conversely, a study conducted by Krizhevsky et al. (2012) that employed a convolutional layer CNN with five convolutional layers and the preprocessed spectrogram as input demonstrated a significantly improved accuracy rate of 84% when applied to the GTZAN dataset. Sènac et al. (2017) utilized CNN to analyze the eight music features spanning three primary dimensions of music—timbre, tonality, and dynamics—on the identical dataset as before (Mouret, 2016; Sènac et al., 2017). Consequently, the CNN model that underwent training attained a maximum accuracy rate of 91%.

Ahmed et al. (2024) presented a deep transfer learning convolutional neural network framework designed to classify air temperature levels—high and low—based on human clothing images. The proposed framework utilizes DeepLabV3+ for semantic segmentation and several pretrained models, including DenseNet121, ResNet101, VGG16/19, BiT, and Vision Transformer (ViT), for classification. The study introduced the Human Clothing Image Dataset (HCID), containing 10,000 labeled images. DenseNet121 achieved the best performance with an accuracy of 98.13%, demonstrating the feasibility of using visual clothing cues for temperature recognition and environmental monitoring applications. In their work, Saadi et al. (2025) enhanced the performance of convolutional neural networks (CNNs) for content-based image retrieval (CBIR) using the COREL image dataset. The proposed approach consists of an offline training phase for feature extraction via a CNN flattening layer and an online retrieval phase employing query-by-image (QBI) comparison using Hamming distance to locate similar images. The system achieved 97.94% accuracy in image classification and 98.94% accuracy in retrieval tasks, demonstrating improved precision and reduced computational time compared to traditional CBIR systems. A study by Almazaydeh et al. (2022) addresses the need for better-categorized online Arabic music content by classifying Arabic music

genres using Deep Convolutional Neural Networks (CNNs) (Hussein et al., 2024). The methodology entails the utilization of Short-Time Fast Fourier Transformation (STFT) to convert audio into spectrograms, feature extraction through the application of Mel Frequency Cepstral Coefficients (MFCC), and a comparative analysis of multiple CNN architectures, including AlexNet, LeNet5, VGG, LSTM-CNN, and ResNet-50. Out of the options considered, AlexNet exhibited the highest level of performance, attaining a classification accuracy of 96% in total. This discovery underscores the efficacy of deep learning methodologies in the automated categorization of Arabic music genres, signifying a substantial progression in retrieving music-related data.

Balaji et al. (2023) put forth a Convolutional Neural Network (CNN) architecture (Fahim et al., 2025) as a means of dysarthria detection in patients with speech impairment. The study employed the TORGO database, which comprised 1080 audio recordings from survivors of stroke and healthy individuals. In order to enhance the robustness of the model and replicate real-world conditions, noise was introduced into these recordings. Identifying dysarthric speech patterns utilizing sophisticated speech feature analysis, such as MFCCs, spectral centroids, and zero crossing rates, was the primary objective of the dataset. Training with 392 audio files, the CNN model was validated with 49 and evaluated with 50. The results revealed a noteworthy detection accuracy of 93.87%, signifying the model's efficacy in promptly identifying dysarthria, a critical factor in facilitating therapeutic interventions. This system is an initial stride in the direction of creating a dependable, automated instrument for the diagnosis and treatment of dysarthria. Bhushan et al. (2021) devised a technique for identifying and rectifying delayed speech through the utilization of Convolutional Neural Networks (CNN) and Weighted Mel Frequency Cepstral Coefficients (MFCC) to extract features. Various speech samples in .wav format were extracted from the UCLASS stuttering dataset to analyze and validate the system. Of the 490 audio files in this set, 392 were utilized for training purposes, 49 were for validation, and 50 were for testing with custom datasets. This study aimed to identify and eliminate speech disfluencies, including prolongations, mute pauses, and repetitions, to produce a corrected text sequence from a signal of interrupted speech. Their approach yielded an optimal accuracy rate of 89%, showcasing its efficacy in enhancing speech recognition (Alqudah et al., 2023) capabilities for users who experience stuttering difficulties. Table 1 below provides a comparative analysis of multiple studies investigating music genre classification (MGC). Support vector machines (SVMs), Hidden Markov Models (HMMs), and advanced deep learning techniques like Convolutional Neural Networks (CNNs) are just a few of the approaches that are highlighted.

Table 1
Comparative Analysis of Multiple Studies Investigating Music Genre Classification (MGC)

Study Reference	Technique Used	Datasets Used	Classification Accuracy
Chai & Vercoe (2001)	Hidden Markov Model (HMM)	Differentiating Irish, German, and Austrian folk music	77%
Xu et al. (2005)	Support Vector Machines (SVM)	-	Superior to HMM
Tzanetakis & Cook (2000)	Genetic Algorithm, SVM, NB, KNN	GTZAN dataset	80.1% (SVM)
Song & Zhang (2008)	SVM	GTZAN and Brazilian Music Dataset (BMD)	86.11% (BMD)
Allamy & Koerich (2021)	1D CNN	GTZAN dataset	80.93%
Sènac et al. (2017)	CNN with 5 convolutional layers	GTZAN dataset	84% using preprocessed spectrogram
Mouret (2016); Sènac et al. (2017)	CNN	GTZAN dataset	91% analyzing timbre, tonality, and dynamics
Mahendran et al. (2023)	CNN with speech features (MFCC, zero crossing rates, spectral centroids, spectral roll off)	TORGO speech signal database	93.87%

3. Materials and methods

The approach shown in Fig. 1 is covered in this part, including the phases of data generation, audio pre-processing, feature extraction, training, and testing.

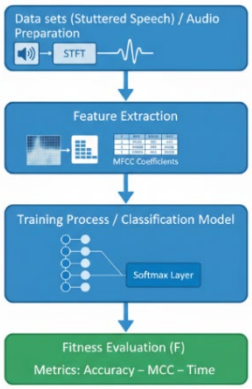

Fig. 1. General Structure

3.1 Datasets (Stuttered Speech) Construction

A multitude of audio samples are extracted to create the vast corpus that makes up the University College London Archive of Stuttered Speech (UCLASS) dataset. As Table 2 illustrates, the dataset is composed of two distinct genre classifications. 807 audio tracks, each lasting 30 seconds, make up the UCLASS. Each audio track is saved as a 491 MB WAV file.

Table 2
Dataset From "University College London's Archive of Stuttered Speech"

Audio	Number of clips
Normal Speech	71
Stuttered Speech	736

The annotations on the UCLASS dataset are comprehensive. "University College London's Archive of Stuttered Speech" (UClass, n.d.) is the title of the UCLASS on the Kaggle repository, where this dataset is presented for free to the research community. The dataset comprises the following folders:

- Genres original—A compilation of 807 audio files from two genres, each lasting 30 seconds.
- Original images—A graphical depiction accompanying each audio file.
- One JSON file—Contains audio file characteristics. The JSON file in question is divided into three distinct sections:
    - "Mapping" includes the genre of music.
    - The labels were numerically represented for each genre [0–4].
    - The vector MFCC is represented by the value 13, comprising the features extracted using MFCC.

3.2 WAV File

The first step in implementing the process is to convert WAV files to spectrograms. In 1991, IBM and Microsoft launched WAVs. They are a waveform audio file type used to store uncompressed, high-fidelity sound recordings. Fig. 2 and Fig. 3 illustrate the waveforms of two UCLASS samples, namely Stuttered speech and Normal speech, with the sample index on the x-axis and the amplitude on the y-axis. In this instance, the visual depictions of both genres unequivocally demonstrate the challenge of differentiating between them; therefore, the ability to anticipate the genre through visual inspection must be made apparent.

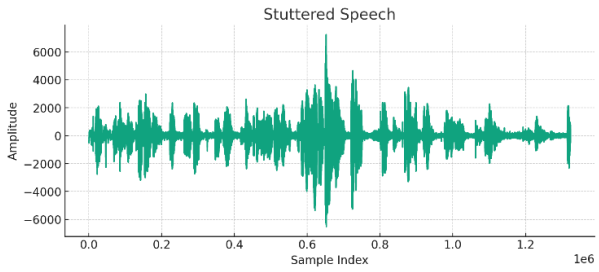

Fig. 2. Waveform of Stuttered Speech

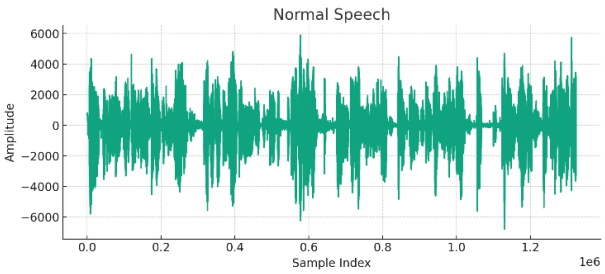

Fig. 3. Waveform of the Normal speech

1. STFT

STFT is the abbreviation for Short Time Fourier Transform. The temporal evolution of the frequency content is illustrated by implementing a DFT algorithm that applies a sequence of windows to the signal. The resultant DFTs are subsequently compiled into a spectrogram, visually represented in Figure 4. The following parameters were utilized in this endeavor to generate the spectrogram:

- 22050 is the sampling rate (SR).
- 2048 is the window size (n_fft).
- Hop_size equals 512.

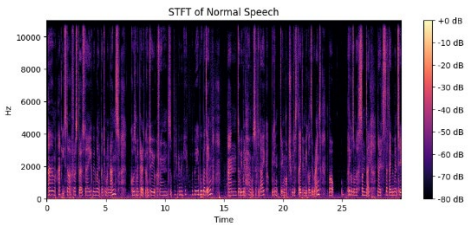

Fig. 4. The resultant DFTs are subsequently compiled into a spectrogram.

2. MFCCs

Mel-Frequency Cepstral Coefficients (MFCCs) were first proposed by Davis and Mermelstein (1980) in the 1990s. Its expedited extraction technique distinguishes it from other cutting-edge feature extraction methods like Linear Prediction Coefficients (LPC) and Perceptual Linear Prediction (PLP). As a result, it has gained significant popularity. By summarizing the frequency distribution across the entire window size, the MFCC enables an examination of the time and frequency attributes of the sound. To commence this study, the audio signal is subjected to the following operations: first, the n_fft 2048 and hop_size 512 STFT is applied; second, the powers' logarithms are calculated at domain frequencies; second, the powers are mapped onto the Mel-Scale using triangle overlapping windows; and third, the MFCCs are obtained by applying the Discrete Cosine Transform (Gupta et al., 2013). Twelve thousand six hundred fifty-six audio characteristics are extracted using the LibROSA application (McFee et al., 2015).

3. CNNs

CNNs, a subtype of deep neural networks, find widespread applicability in computer vision tasks. Additionally, CNNs may be used for a variety of audio analysis applications because the 2D CNNs' design, which accepts spectrograms as input, can process audio data following MFCC transformation. In the same way that the architecture of two-dimensional CNNs employed in image processing tasks enables the processing of spectrograms as images, variations in musical patterns also permit this. A spectrogram, a two-dimensional visual depiction of the temporal evolution of the spectrum of audio frequencies, is produced by convolving the input with filters that discern critical attributes to coincide with the output. As a consequence, CNN acquires predominantly hierarchical features via convolutions. The output of the initial convolution layer is a dot product formed by convolving the pixels comprising the entire image with the filter. When this feature map reaches the last node of the network, it includes incredibly comprehensive general information on the contents of the picture. From there, it is passed on to the next layer, which creates an endless number of feature maps. The filters' weights, or the numerical values inside them, represent the parameters that were taught during the training stage. An activation function and a pooling layer sit between each convolution layer of a CNN, which is used to find hierarchical features. The activation functions utilize the backpropagation method to compute the error, which is disseminated throughout the network to modify the filter weights. The Rectified Linear Unit (ReLU) function is the most frequently employed activation function. Pooling layers are usually an extra component of a CNN; max pooling is the most common technique employed in pooling. The purpose of pooling is to minimize the number of parameters and simplify the network (Lin & Shen, 2018).

As we have just explained, this is the basic CNN architecture (Kumar et al., 2018). Since the first CNN architecture, LeNet5, created by LeCun et al. (1998), several others have been created. The same was done by Krizhevsky et al. (2012) with AlexNet (Krizhevsky et al., 2012). The deep learning network was used to classify 1.2 million images into 1,000 unique classes in the ImageNet LSVRC-2010 competition, the top object identification competition, in 2012. Simonyan and Zisserman (2014) created VGG, which is in line with GPU developments. ResNet was later created in 2016 by He et al. (2016). At the moment, the majority of state-of-the-art architectures (Adolf et al., 2016) show similar performance, with very few use cases where they outperform.

4. LeNet5

Since LeNet is among the first deep learning designs, it belongs to the most popular class of neural network architectures. LeNet5 is composed of five convolutional and pooling layers that alternate, with two fully connected layers at the end. It utilizes average pooling for feature space down-sampling, ReLU and softmax activation between layers, and convolution to maintain the spatial orientation of features. Table 3 displays the LeNet5 model's summary. The overview shows the total number of layers, the input size for each layer, the activation functions used, and other parameters.

Table 3
The LeNet5 Architecture and the Model Summary

Layer	Feature map	Size	Kernel size	Stride	Activation
Input	Audio	1	(132, 13, 1)	-	-
1	Convolution	6	(128, 11, 6)	(3, 3)	-
2	Average pooling 2D	6	(64, 5, 6)	-	-
3	Convolution	16	(62, 3, 16)	(3, 3)	-
4	Average pooling 2D	16	(31, 1, 16)	-	-
5	Convolution	120	(120)	-	-
6	FC	-	(84)	-	-
Output	FC	-	(10)	-	-

5. AlexNet

The first deep neural network of the twenty-first century was called AlexNet. The more sophisticated LeNet5 model is the one that emerged victorious in the highly regarded ImageNet LSVRC-2010 object identification competition, successfully classifying over a million photos into 1000 distinct classes. The network did well in the competition thanks to advancements

in GPU technology, processing hardware, and the availability of large datasets. There are eight learned levels in the AlexNet architecture. Three completely linked layers are layered after five convolution layers. A nonlinear activation function (ReLU) comes after a convolutional filter in each convolutional layer. The first and second convolutional layers' learning is quantitatively stabilized using max pooling and normalization techniques, respectively, while shift invariance is improved. The third, fourth, and fifth convolutional layers are not connected to one another by any pooling or normalizing layers. Table 4 shows the summary of the AlexNet model. The summary shows the total number of layers, the amount of input for each layer, the activation functions used, and any extra parameters.

Table 4
The Model Overview and the AlexNet Architecture

Layer	Feature map	Size	Kernel size	Stride	Activation
Input	Audio	-	-	-	-
1	Convolution	96	(33, 4, 96)	(11,11)	(4,4)
2	Max pool	96	(33, 4, 96)	(3,3)	(2,2)
3	Convolution	256	(17, 2, 256)	(5,5)	(2, 2)
4	Max pool	256	(9, 1, 256)	(3, 3)	(2, 2)
5	Convolution	384	(9, 1, 384)	(3, 3)	(1, 1)
6	Convolution	384	(9, 1, 384)	(3, 3)	(1, 1)
7	Convolution	256	(9, 1, 384)	(3, 3)	(1, 1)
8	Max pool	256	(5, 1, 256)	(3, 3)	(2, 2)
9	FC	-	4096	-	-
10	FC	-	4096	-	-
11	FC	-	1000	-	-

6. VGG

The Oxford Visual Geometry Group created the VGG model in 2014 to address depth, which is another important aspect of convolution architecture design. While AlexNet has eight levels, VGG would have eleven to nineteen layers. Other architectural criteria were established with this goal in mind. Simultaneously, depth was progressively raised by adding more convolutional layers, made feasible by the comparatively modest convolution filters present at every layer level. The convolution layers are stacked after the spectrograms are processed. The lowest size needed to convey the concepts of left-right, up-down, and center is 3 × 3, which is the receptive field of the filters. As a result, the parameter significantly drops. Spatial pooling is also performed by five max pooling layers, which are positioned after part of the convolutional layers. Table 5 provides a summary of the VGG model. The overview displays all the layers, the input size of each layer, the activation functions that were applied, and any additional parameters.

Table 5
Model Overview and VGG Architecture

Layer	Feature map	Size	Kernel size	Stride	Activation
Input	Audio	1	(132, 13, 1)	-	-
1	2×convolution	64	(130, 13, 64)	(3, 3)	(2, 2)
2	1×max pool	64	(65, 6, 64)	(3, 3)	-
3	2×convolution	128	(65, 6, 128)	(3, 3)	-
4	1× max pool	128	(32, 3, 128)	(3, 3)	(2, 2)
5	3×convolution	256	(32, 3, 256)	(3, 3)	-
6	1×max pool	256	(16, 1, 256)	(3, 3)	(2, 2)
7	3×convolution	512	(16, 1, 512)	(3, 3)	-
8	1×max pool	512	(8, 1, 512)	(3, 3)	(2, 2)
9	3×convolution	512	(8, 1, 512)	(3, 3)	-
10	1×maxpool	512	(4, 1, 512)	(3, 3)	(2, 2)
11	FC	512	4096	-	-
12	FC	-	4096	-	-
Output	FC	-	10	-	-

7. ResNet-50

While the ResNet may be trained to a maximum depth of 152 layers in the network, earlier models typically had depths of 16 or 30 layers.

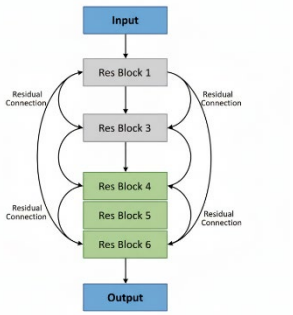

Fig. 5. The flowchart for ResNet

Compared to the VGG nets, this is eight times greater. A CNN with 50 layers and residual connections is called ResNet-50. As such, the layers are construed not as learning unreferenced functions but as learning residual functions regarding the layer inputs. Fig. 5 shows a simple ResNet flow chart with the residual connection—the building block that employs skip connections to bypass certain layers—illustrated. Skip connections are mostly used to solve the Degradation (accuracy saturation) problem and prevent fading gradients. These models are called Highway Nets; moreover, an extra weight matrix may be employed.

Table 6
The Model Summary and the ResNet Architecture

Layer Type	Output Shape	Number of Filters	Kernel Size	Stride
Input	224×224×3	-	-	-
Convolutional	112×112×64	64	7 × 7	2
Residual Block	112×112×64	64	3 × 3	1
Residual Block	112×112×64	64	3 × 3	1
Down sampling	56×56×128	128	3 × 3	2
Residual Block	56×56×128	128	3 × 3	1
Residual Block	56×56×128	128	3 × 3	1
Down sampling	28×28×256	256	3 × 3	2
Residual Block	28×28×256	256	3 × 3	1
Residual Block	28×28×256	256	3 × 3	1
Down sampling	14×14×512	512	3 × 3	2
Residual Block	14×14×512	512	3 × 3	1
Residual Block	14×14×512	512	3 × 3	1
Global Average Pooling	1×1×512	-	-	-
Fully Connected	1×1×1000	-	-	-

4. Results

One of the main objectives of the study is to evaluate the trained models' performance, as described in Section III, using our freshly created UCLASS dataset, which consists of 807 audio recordings with 30-second segments each, recorded as WAV audio files at a sample rate of 22050 Hz. The audio genres of Normal Speech and Stuttered Speech were used to categorize the samples. The training set of the UCLASS dataset was divided into 75% and the testing set into 25%. The experiment was conducted using Python 3 on a GPU server (Kaggle notebook) to develop and assess our technique (Python Software Foundation, n.d.). The deep CNN models were trained using audio signals, which are used to compute MFCC features. In this paper, ResNet50, VGG16, LeNet, AlexNet, and CNN architectures were used, and classification performance metrics, such as accuracy, recall, precision, and F-score, were acquired for each of them. The four possible outcomes that these metrics are based on are false positive (FP), true negative (TN), true positive (TP), and false negative (FN).

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{1}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{2}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{3}$$

$$F1\ Score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)} \tag{4}$$

An empirical experiment was conducted to evaluate the efficacy of CNN models by varying epochs and learning rate parameters. The learning rate determines whether a CNN model learns a problem swiftly or slowly, and an epoch is one iteration through the entire training dataset. Determining the optimal number of epochs for a network is, in fact, a skill in machine learning; thus, two epoch values (90 and 100) and three learning rates (0.01, 0.001, and 0.0001) were implemented. Table 7 presents the effects of different learning rates and epochs on the AlexNet model's performance, as evaluated by Accuracy, F1-Score, Precision, and Recall. It is worth mentioning that the highest accuracy (95.638%) and F1-Score (95.731%) are obtained with a learning rate of 0.001 and 90 epochs, suggesting that these parameters define optimal performance. Additionally, the combination of this learning rate and epoch yields the highest precision, highlighting the importance of striking a balance between adjusting the learning rate and the duration of model training to achieve optimal performance.

Table 7
AlexNet Performance Metrics

Learning Rate	Epochs	Accuracy	F1-Score	Precision	Recall	TP	FP	FN	TN
0.0001	90	0.933226	0.933937	0.934721	0.933226	107	58	66	1626
0.0001	100	0.955304	0.950632	0.953179	0.955304	93	72	11	1681
0.001	90	0.815293	0.851527	0.935613	0.815293	159	6	337	1355
0.001	100	0.950996	0.94523	0.948199	0.950996	86	79	12	1680
0.01	90	0.951535	0.947749	0.94792	0.951535	96	69	21	1671
0.01	100	0.934841	0.919387	0.934367	0.934841	51	117	4	1685

Table 8 presents the performance of the LeNet model at various learning rates and epochs. It demonstrates a particularly high accuracy of 90.199% when the learning rate is set to 0.0001 and the number of epochs is 100. Nevertheless, the model attains an optimal trade-off between Precision and Recall while maintaining the same learning rate and number of epochs. This indicates that maximizing model effectiveness, specifically in terms of accuracy and F1-Score, requires a minimal learning rate and prolonged training periods.

Table 8
LeNet Performance Metrics

Learning Rate	Epochs	Accuracy	F1-Score	Precision	Recall	TP	FP	TN	FN
0.0001	90	0.901454	0.877882	0.874133	0.901454	1644	154	30	29
0.0001	100	0.901992	0.891105	0.885774	0.901992	1618	127	57	55
0.001	90	0.878837	0.864658	0.854513	0.878837	1598	150	34	75
0.001	100	0.900915	0.887048	0.881314	0.900915	1624	135	49	49
0.01	90	0.900915	0.853956	0.811649	0.900915	1673	184	0	0
0.01	100	0.900915	0.853956	0.811649	0.900915	1673	184	0	0

Table 9 presents the results of the ResNet-50 models. The findings suggest that achieving maximum accuracy (93.915%) and F1-Score (93.334%) requires a learning rate of 0.01 across 100 epochs. The findings indicate that increasing the learning rate and training duration substantially improves the accuracy and recall of the ResNet-50 models, thus establishing a viable approach to optimizing their performance.

Table 9
ResNet-50 Performance Metrics

Learning Rate	Epochs	Accuracy	F1-Score	Precision	Recall	TP	FP	TN	FN
0.0001	90	0.928918	0.918495	0.920783	0.928918	1653	110	72	22
0.0001	100	0.921917	0.919987	0.918458	0.921917	1612	82	100	63
0.001	90	0.902531	0.909586	0.920784	0.902531	1548	54	128	127
0.001	100	0.936995	0.931216	0.931078	0.936995	1647	89	93	28
0.01	90	0.924071	0.904639	0.921405	0.924071	1669	135	47	6
0.01	100	0.939149	0.933343	0.93374	0.939149	1650	88	94	25

The VGG16 outcomes demonstrate a clear trend: the model attains remarkable test accuracy and F1-score of 97.307% and 97.248%, respectively, with a learning rate of 0.0001 and 100 epochs. The presented data illustrate that to optimize the efficacy of the VGG16 model, particularly regarding accuracy and F1-score, it is critical to employ lengthy training periods and low learning rates throughout the evaluated parameters.

Table 10
VGG16 Performance Metrics

Learning Rate	Epochs	Accuracy	F1-Score	Precision	Recall	TP	FP	TN	FN
0.0001	90	0.962843	0.963567	0.964715	0.962843	1610	25	178	44
0.0001	100	0.972536	0.972324	0.972179	0.972536	1632	29	174	22
0.001	90	0.890684	0.839186	0.793318	0.890684	1654	203	0	0
0.001	100	0.890684	0.839186	0.793318	0.890684	1654	203	0	0
0.01	90	0.890684	0.839186	0.793318	0.890684	1654	203	0	0
0.01	100	0.890684	0.839186	0.793318	0.890684	1654	203	0	0

5. Conclusion

The performance of four trained CNN models (ResNet50, VGG16, LeNet, and AlexNet) has been evaluated in terms of accuracy, recall, precision, and F-score as classification measures. The overall results showed that all four models demonstrated high performance in speech classification. The VGG16 demonstrates the highest accuracy of 97.307% with a learning rate of 0.0001 and 100 epochs, followed by the AlexNet model's accuracy of 95.638% with a learning rate of 0.001 and 90 epochs, while the ResNet-50 and the LeNet model showed less accuracy of 93.915% and 90.199%, respectively. For maximizing LeNet model effectiveness, a minimal learning rate and prolonged training periods are required. On the other hand, increasing the learning rate and training duration substantially improves the accuracy and recall of the ResNet-50 models.

Acknowledgement

The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through the Large Research Project under grant number RGP.2/435/46.

References

Adolf, R., Rama, S., Reagen, B., Wei, G. Y., & Brooks, D. (2016). Fathom: Reference workloads for modern deep learning methods. *2016 IEEE International Symposium on Workload Characterization (IISWC)*. IEEE. https://doi.org/10.1109/IISWC.2016.7581275

Ahmed, M., Zhang, X., Shen, Y., Ali, N., Flah, A., Kanan, M., Alsharef, M., & Ghoneim, S. S. M. (2024). A deep transfer learning-based convolutional neural network framework for air temperature classification using human clothing images. *Scientific Reports, 14*, 31658. https://doi.org/10.1038/s41598-024-80657-y

Akhtar, S., Aftab, S., Kousar, S., Rehman, A., Ahmad, M., & Saeed, A. Q. (2024). Performance analysis of training optimizers on diabetic retinopathy detection using SP-Net CNN. *2024 International Conference on Decision Aid Sciences and Applications (DASA)*, 1–5. https://doi.org/10.1109/dasa63652.2024.10836543

Al-Anazi, R. G., Alqahtani, H., Alzaidi, M. S. A., Alanazi, M. H., Sultan, H. A., Alrowaily, A. F., Aljabri, J., & Alqudah, A. (2024). Multi-class automated speech language recognition using natural language processing with optimal deep learning model. *Fractals*. https://doi.org/10.1142/s0218348x25400213

Allamy, S., & Koerich, A. (2021). 1D CNN architectures for music genre classification. *arXiv preprint arXiv:2105.07302*. https://arxiv.org/abs/2105.07302

Almazaydeh, L., Atiewi, S., Tawil, A. A., & Elleithy, K. (2022). Arabic music genre classification using deep convolutional neural networks (CNNs). *Computers, Materials & Continua, 72*(3), 5443–5458. https://doi.org/10.32604/cmc.2022.025526

Alqudah, A. A. M., Alshraideh, M. A. M., Sharieh, A. A. S., Abushariah, M. A. M., & Quiam, F. M. D. (2023). CNN. *2023 International Conference on Information Technology (ICIT)*, 667–673. https://doi.org/10.1109/ICIT58056.2023.10225965

AlZu'bi, S., Alqudah, A., Quiam, F., & Mughaid, A. (2024). Human Voice in Health Data: NLP for detecting synthetic voices in medical documentation, social media, and biological literature. SSRN Electronic Journal.https://doi.org/10.2139/ssrn.4757246

Bagci, U., & Erzin, E. (2007). Automatic classification of musical genres using inter-genre similarity. *IEEE Signal Processing Letters, 14*(8), 521–524.

Bergstra, J., Casagrande, N., Erhan, D., Eck, D., & Kegl, B. (2006). Aggregate features and adaboost for music classification. *Machine Learning, 65*(2-3), 473–484. https://doi.org/10.1007/s10994-006-8712-3

Bhushan, P., Vani, H. Y., Shivkumar, D. K., & Sreeraksha, M. R. (2021). Stuttered speech recognition using convolutional neural networks. *International Journal of Engineering Research & Technology (IJERT), 9*(12). https://www.ijert.org

Chai, W., & Vercoe, B. (2001). Folk music classification using hidden Markov models. *Proceedings of International Conference on Artificial Intelligence, 6*(6), 1–6.

Cheng, H., Chang, C., & Kuo, C. (2020). Convolutional neural networks approach for music genre classification. *2020 International Symposium on Computer, Consumer and Control (IS3C)*, 399–403. https://doi.org/10.1109/IS3C50286.2020.00111

Choi, K., Fazekas, G., & Sandler, M. (2016). Automatic tagging using deep convolutional neural networks. *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR 2016)*, 805–811.

Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 28*(4), 357–366. https://doi.org/10.1109/TASSP.1980.1163420

de Sousa, J., Pereira, E., & Veloso, L. (2016). A robust music genre classification approach for global and regional music datasets evaluation. *IEEE International Conference on Digital Signal Processing (DSP)*, 109–113. https://doi.org/10.1109/ICDSP.2016.7868522

Defferrard, M., Benzi, K., Vandergheynst, P., & Bresson, X. (2017). FMA: A dataset for music analysis. *18th International Society for Music Information Retrieval Conference*, 316–323.

Fahim, A., Osman, A. M., Tarek, Z., & Elshewey, A. M. (2025). A hybrid deep learning framework based on CNN-GRU-TabNet for the predictive modeling of COVID-19 mortality. *Engineering Technology & Applied Science Research, 15*(5), 28057–28062. https://doi.org/1Convolutional Neural Network (CNN)0.48084/etasr.13910

Girsang, A., Manalu, A., & Huang, K. (2019). Feature selection for musical genre classification using a genetic algorithm. *Advances in Science, Technology and Engineering Systems, 4*(2), 162–169. https://doi.org/10.25046/aj040220

Gupta, S., Jaafar, J., Ahmad, W. F. W., & Bansal, A. (2013). Feature extraction using MFCC. *Signal & Image Processing: An International Journal, 4*(4), 101–108. https://doi.org/10.5121/sipij.2013.4408

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Hussein, A. M., Sharifai, A. G., Alia, O. M., Abualigah, L., Almotairi, K. H., Abujayyab, S. K. M., & Gandomi, A. H. (2024). Auto-detection of the coronavirus disease by using deep convolutional neural networks and X-ray photographs. *Scientific Reports, 14*(1), 534. https://doi.org/10.1038/s41598-023-47038-3

Jr, C., Koerich, A., & Kaestner, C. (2008). The latin music database. *International Society for Music Information Retrieval Conference*, 451–456.

Kim, T., Lee, J., & Nam, J. (2018). Sample-level CNN architectures for music auto-tagging using raw waveforms. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 366–370. https://doi.org/10.1109/ICASSP.2018.8461671

Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. *Proceedings of the 25th International Conference on Neural Information Processing Systems, 1*, 1097–1105.

Kumar, A., Rajpal, A., & Rathore, D. (2018). Genre classification using feature extraction and deep learning techniques. *2018 10th International Conference on Knowledge and Systems Engineering (KSE)*, 175–180. Hong Kong.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324. https://doi.org/10.1109/5.726791

Li, T., & Ogihara, M. (2006). Toward intelligent music information retrieval. *IEEE Transactions on Multimedia, 8*(3), 564–574. https://doi.org/10.1109/TMM.2006.870737

Li, T., Chan, A., & Chun, A. (2010). Automatic musical pattern feature extraction using convolutional neural network. Proceedings of the International Multi-Conference of Engineers and Computer Scientists (IMECS), 1–6.

Lin, G., & Shen, W. (2018). Research on convolutional neural network based on improved Relu piecewise activation function. *Procedia Computer Science, 131*, 977–984. https://doi.org/10.1016/j.procs.2018.04.239

Mahendran, M., Visalakshi, R., & Balaji, S. (2023). Dysarthria detection using convolution neural network. *Measurement: Sensors, 30*, 100913. https://doi.org/10.1016/j.measen.2023.100913

McFee, B., Raffel, C., Liang, D., Ellis, D. P. W., McVicar, M., Battenberg, E., & Nieto, O. (2015). Librosa: Audio and music signal analysis in python. *Proceedings of the 14th Python in Science Conference*, 18–25.

McKay, C., & Fujinaga, I. (2005). Automatic music classification and the importance of instrument identification. Proceedings of the Conference on Interdisciplinary Musicology, 1–10. Canada.

Mouret, F. (2016). *Personalized music recommendation based on audio features* [Master's thesis, INP ENSEEIHT]. Toulouse, France.

Python Software Foundation. (n.d.). *Python 3.9.5*. https://www.python.org/downloads/release/python-395/

Saadi, Z. M., Sadiq, A. T., Akif, O. Z., & El-Kenawy, E. M. (2025). Enhancing convolutional neural network for image retrieval. *Journal of Intelligent Systems and Internet of Things, 14*(2), 140–152. https://doi.org/10.54216/JISIoT.140212

Scaringella, N., Zoia, G., & Mlynek, D. (2006). Automatic genre classification of music content: A survey. *IEEE Signal Processing Magazine, 23*(2), 133–141. https://doi.org/10.1109/MSP.2006.1598089

Sènac, C., Pellegrini, T., Mouret, F., & Pinquier, J. (2017). Music feature maps with convolutional neural networks for music genre classification. *International Workshop on Content-Based Multimedia Indexing (CBMI)*, 1–5. https://doi.org/10.1145/3095713.3095716

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *Proceedings of the International Conference on Learning Representations*.

Song, Y., & Zhang, C. (2008). Content-based information fusion for semi-supervised music genre classification. *IEEE Transactions on Multimedia, 10*(1), 145–152. https://doi.org/10.1109/TMM.2007.911304

Turnbull, D., & Elkan, C. (2005). Fast recognition of musical genres using RBF networks. *IEEE Transactions on Knowledge and Data Engineering, 17*(4), 580–584. https://doi.org/10.1109/TKDE.2005.62

Tzanetakis, G., & Cook, P. (2000). MARSYAS: A framework for audio analysis. *Organised Sound, 4*(3), 169–175. https://doi.org/10.1017/S1355771800003071

UClass. (n.d.). *UCL 1999-2005*. https://www.uclass.psychol.ucl.ac.uk/

Xu, C., Maddage, N., & Shao, X. (2005). Automatic music classification and summarization. *IEEE Transactions on Speech and Audio Processing, 13*(3), 441–450. https://doi.org/10.1109/TSA.2003.822599

Zena, Z., Sadiq, A. T., Akif, O. Z., & El-Kenawy, E. M. E. (2024). Enhancing convolutional neural network for image retrieval. *Journal of Intelligent Systems and Internet of Things*, *14*(2), 140–152. https://doi.org/10.54216/jisiot.140212

© 2026 by the authors; licensee Growing Science, Canada. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (http://creativecommons.org/licenses/by/4.0/).