Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Real-Time Cough Detection and Classification of COVID-19 Using LSTM-Based Sound Separation and Lightweight CNN Models

[1]Shroog Abdulmatlub Alzanbagi, [2]Adil Ahmed, [3]Saim Rasheed

[1]Department of Information Technology, King Abdulaziz Universty

Jeddah, Saudi Arabia

Email: ssalahalzanbaki@stu.kau.edu.sa

[2]Department of Information Technology, King Abdulaziz Universty

Jeddah, Saudi Arabia

Email: aahmad@kau.edu.sa

[3]Department of Information Technology, King Abdulaziz Universty

Jeddah, Saudi Arabia

Email: srahmed@kau.edu.sa

ARTICLE INFO

Received: 25 Dec 2024

Revised: 15 Feb 2025

Accepted: 25 Feb 2025

ABSTRACT

COVID-19, a respiratory disease, caused severe human, social, and economic loss worldwide. Early-stage diagnosis of COVID-19 can help to mitigate its spread and health complications. However, existing diagnosis methods involve high costs and can put healthcare professionals at risk of infection. To address these challenges, this paper presents a lightweight sound separation based on Long Short-Term Memory (LSTM) and lightweight Convolutional Neural Network (CNN) model for real-time detection and classification of COVID19 based on cough sounds. The proposed approach does not require the in-person presence of patients, eliminating the risk of spreading the virus. Background noises in cough sounds pose a significant challenge to classification accuracy. This study acquires cough sound data from six credible sources, removes background noises from them using a deep learning technique, and finally includes 1,886 COVID-19-positive and 1,757 COVID-19-negative samples in the dataset. The performance of deep learning models i.e., MobileNetV2, MobileNetV3 Small, and EfficientNet-lite-0 is evaluated using the confusion matrix. Results indicate that MobileNetV3 Small outperforms all other models with an accuracy of 99%, making it the best choice for real-time detection and classification of cough-based COVID-19.

Keywords: Real-time; cough; COVID-19; deep learning; sound separation; LSTM; Lightweight CNN; classification

I. Introduction

COVID-19 is a respiratory disease caused by the SARS-CoV – 2 coronavirus. Its basic symptoms include sore throat, cough, fever, chest pain, breathing difficulty, and loss of taste and smell etc. It was declared a global pandemic by the World Health Organization (WHO) in 2020 and continues to have widespread personal, societal, and economic impacts on the world [1] .According to WHO, as of March 2025, 7.1 million people have died due to COVID-19 globally [2]. The symptoms of COVID-19 are similar to

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

symptoms of pneumonia. Therefore, in some cases, people don't follow the protocol recommended for COVID-19 patients. This results in health complications and the spread of the virus among healthy individuals. Thus, its early detection becomes crucial.

Early-stage diagnosis of COVID-19 can help treat infected patients and control its spread. Reverse Transcriptase Polymerase Chain Reaction (RT-PCR) and CT Scan are the popular methods for diagnosing COVID-19 [3].

The RT-PCR test is considered the gold standard for diagnosing COVID-19. It offers high diagnostic accuracy [4]. However, it has some limitations, such as the need for specialized personnel and equipment, exposure to diluted environments, and lack of sensitivity. The requirement for specialized personnel and equipment makes it impractical for rapid and large-scale screening [3]. Exposure of healthcare professionals to a diluted environment can put them at risk of COVID-19, and lack of sensitivity can result in misdiagnosis.

CT Scan is an effective method for diagnosing COVID19. It provides a clear view of the lung's health, which helps understand complications caused by the virus. However, it also has some limitations, such as exposure to high radiation, high cost, and dependency on PCR testing. CT Scan involves a high radiation dose, which might be dangerous for the health of kids and pregnant women. Apart from that, sometimes multiple CT Scans are required for some patients, which put them at risk of long-term health complications. The CT Scan involves a high cost, which makes it unaffordable for ordinary people. Pneumonia and other viruses also cause lung damage. The findings of the CT Scan report will remain the same for all types of viruses. Therefore, PCR testing is used for diagnosing disease.

Researchers are working to propose accurate, cost effective, practical, and easy-to-use COVID-19 screening tools that can overcome the limitations of existing diagnostic methods. Deep learning, a subset of machine learning. Deep learning models contain multiple layers of neurons, which help them learn complex and large datasets. Therefore, for large datasets, deep learning models achieve higher accuracy than machine learning models. Deep learning models have wide application for the classification of COVID-19.

Numerous deep learning models use X-rays, CT scans, cough, and other types of data, collected through wearable devices to classify COVID-19 with great accuracy. Respiratory diseases affect the acoustic features of cough sounds [5], and coughing sounds have different bandwidths and qualities than a whole spoken signal. Cough-based COVID-19 detection efficiency of deep learning models can be enhanced by utilizing these features [6].

II. Related Work

The COVID-19 pandemic has notably accelerated the development of machine learning and deep learning models for non-invasive diagnostics, particularly those leveraging cough sounds. While these systems have shown considerable promise. Their performance in real-world environments remains hindered by different challenges such as overlapping audio signals, which can significantly degrade classification accuracy.

This section explores some of the recent advancements in cough-based COVID-19 classification, emphasizing the role of sound separation techniques as a critical solution to the degraded sound problem.

In [7], the researchers introduced a deep Convolutional neural network (DCNN) capable of classifying COVID-19 from various respiratory sounds, including coughs, breaths, and vocalizations. This model employed methods of Gammatone Frequency Cepstral Coefficients (GFCCs), Improved Mel-Frequency Cepstral Coefficients (IMFCCs), and a Denoising Autoencoder (DAE) to extract clean, informative features from raw audio files. Although the system achieved a high classification accuracy of 93% for coughs and 95% when incorporating additional respiratory sounds. Its computational complexity limits its practicality in real time.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The ALCOVID [8] model adopted a two-stage approach encompassing detection and classification. The detection component utilizes a Convolutional Neural Network (CNN) to process Mel-spectrogram images to detect cough events from other sounds. After that, the identified segments are forwarded to cloud-based classifiers for diagnostic evaluation. However, this is vulnerable in noisy environments, often discarding data with substantial interference, potentially excluding crucial diagnostic signals. For classification purposes, ALCOVID integrates three complementary models to enhance diagnostic reliability. The Deep Transfer Learning-based Multi-Class Classifier (DTLMC) distinguishes between COVID-19, pertussis, bronchitis, and healthy cases with 92.64% accuracy, leveraging transfer learning techniques. The Classical Machine Learning-based Multi-Class Classifier (CMLMC) utilizes a Support Vector Machine (SVM) with MelFrequency Cepstral Coefficient (MFCC) features refined through Principal Component Analysis (PCA), and achieved 88.76% accuracy. The Deep Transfer Learning-based Binary Classifier (DTLBC) specifically differentiates COVID-19 coughs, attaining 92.85% accuracy. Importantly, when classifier outputs conflict, the system provides an "inconclusive" result to reduce diagnostic errors. The authors in [9] proposed a lightweight deep learning model tailored for embedded systems to detect COVID-19 through cough analysis. Designed for real-time, on-device deployment, the model prioritizes computational efficiency without sacrificing accuracy and essential attributes for scalable, non-clinical screening during pandemics. The architecture incorporates quadratic convolutional layers and kernel separation techniques, striking a balance between performance and resource consumption. Tested on the Virufy dataset [10], the model achieved a remarkable accuracy of 97.5%. However, its performance under noisy or uncontrolled conditions remained unverified, raising concerns about its generalizability.

The authors in [11] introduced a deep learning framework that transforms cough sounds into scalogram images to capture their time-frequency characteristics. They acquired COUGHVID dataset[19], which contained 1,457 samples (755 COVID-19 positive, 702 healthy). They implemented and evaluated the performance of six DL models, i.e., GoogleNet, ResNet18, ResNet50, ResNet101, MobileNetV2, and NasNetMobile. Preprocessing involved filtering extraneous low and high-frequency noise to enhance signal clarity. Among the tested models, ResNet18 achieved the highest accuracy of 94.9%, with sensitivity and specificity values of 94.44% and 95.37%, respectively.

These limitations indicate a clear need to develop techniques to extract and improve target sounds within complex acoustic environments. Sound Separation (SSep) is an advanced technique that is used as a preprocessing step for extracting distinct audio sources from complex audio signals. Combining SSep with sound event detection (SED) reduces interference and allows the identification of overlapping sounds, which makes a notable improvement in the accuracy and robustness of detection systems. SSep provides clean and identifiable inputs. In the past, this ability was limited to speech and music only, but now is expanded to sound separation. Integrating SSep with the audio processing pipeline shows a promising way forward, especially in real-world scenarios where background interference creates challenges that must be overcome while keeping important diagnostic information accessible [12].

Model compression [13] allows for quick inference, energy efficiency, and environment friendly deployment in real-time settings. Pruning removes unimportant parameters, which reduces memory requirements and speeds up computations. It is crucial for real-time applications that need low latency and high reliability. Standard pruning methods require a series of pruning and retraining cycles. Sparsity constraints lack generalizability, involve high computational costs, and are hyperparameter sensitive [14]. To overcome these limitations, the authors [14] introduced a single-shot pruning technique that uses a connection sensitivity approach to determine the significance of network connections during evaluation and efficient pruning without extensive retraining. Mathematically, the connection sensitivity $S(\theta_q)$ is defined as:

$$S(\theta_q) = \lim_{\epsilon \to 0} \frac{\left| L(\theta_0) - L(\theta_0 + \epsilon \delta_q) \right|}{\epsilon} = \left| \theta_q \frac{\partial L}{\partial \theta_q} \right| \quad (1)$$

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Here θ_q represents the q-th weight in the initial parameter set θ_0, and δ_q is a one-hot vector indicating a perturbation in only the q-th element. This quantifies the impact of each weight on the loss function. Unlike traditional methods, SNIP prunes connections with nominal effect on the loss function prior to training, which ensures computational efficiency and integrity of the model.

Source separation methods separate single sound sources from a combination of audio signals. Here, source means every single original signal in the combined audio signals, and the process that recovers these sources is called Blind Source Separation (BSS). Single channel separation is a complex form of BSS, where only a single combined audio signal is available. This scenario poses a significant challenge due to the entanglement of multiple sources in both the time and frequency domains [15].

Our research focuses on the complexity of real-world acoustic environments, where overlapping signals are common. In such environments, the target sound is masked by the other competing signals, specifically when these signals have the same amplitudes. This significantly hinders the perception and interpretation of the sound of interest. To address this challenge, audio separation is applied to extract individual sources from a complex signal.

III. Methodology

This study aims to optimize lightweight deep learning models for the real-time detection and classification of COVID-19 based on cough sounds. Detection of cough in an acoustic environment remains a great challenge. To address this challenge, we applied a mask-based sound separation model as a preprocessing step to isolate cough sounds. The sound separation models involve high costs. To handle this, we applied pruning techniques to reduce model complexity without degrading sound quality. Then, we implemented three lightweight deep learning models, that is, MobileNetV2, MobileNetV3-Small, and EfficientNet-Lite-0 for cough sound classification and evaluated their performance. Figure 1 illustrates our proposed methodology.

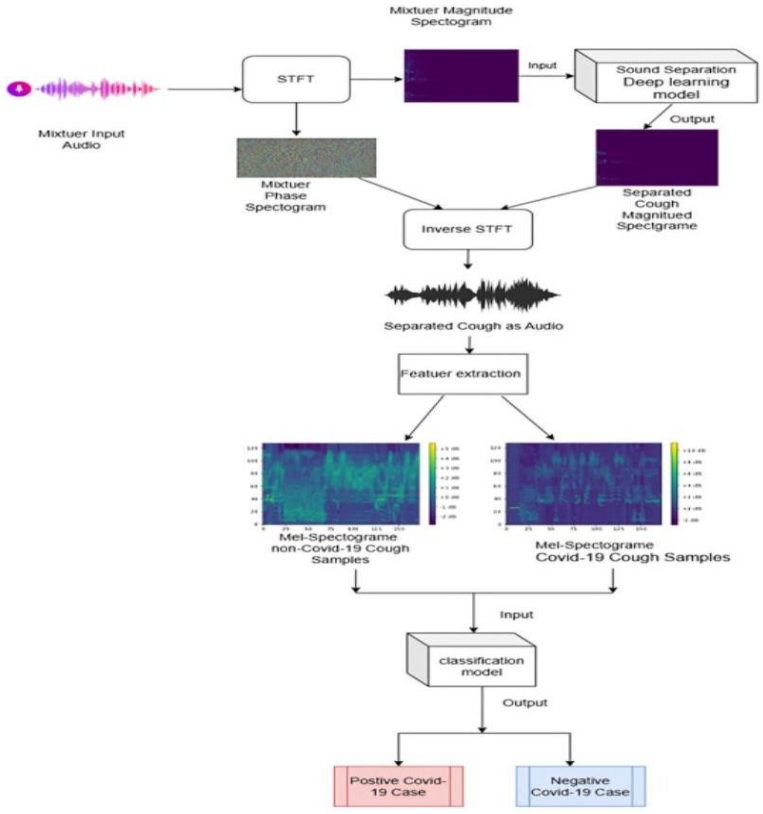


Fig. 1: Proposed Methodology

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

A. Dataset

This study used six datasets that encompass a wide range of COVID-19-related cough sounds, collected in various geographical regions and demographic groups.

Generally, crowdsourced data contains irrelevant samples. A pre-trained eXtreme Gradient Boosting (XGBoost) classifier was employed to shortlist the recordings, retaining only those with a cough probability score above 0.80 [16]. This threshold ensured that the majority of the detected coughs were genuine. Such a balance minimizes bias and errors in cough detection, enhancing the classifier's utility for researchers [16].

IATos Dataset [17] includes audio samples from individuals undergoing COVID-19 testing at public and private healthcare facilities. The collected data contained recordings from 2,821 participants, all of whom were found to be positive or negative via RT-PCR. The sample featured a nearly even gender distribution with 61.1% aged between 21 and 40 . Participants, recruited from febrile emergency units, testing centers, isolation facilities, and private clinics, were asked to submit one cough sample daily for three days. Each sample was collected within three days of the RT-PCR test, ensuring strong temporal alignment with diagnosis.

CCS (COVID-19 Cough Sub-Challenge) [18] was developed via a web and Android app to crowdsource cough and respiratory sound samples globally. Participants self-reported their COVID-19 status, symptoms, demographics, and relevant medical history. It includes both positive and healthy samples, with the latter encompassing healthy individuals and those with other respiratory conditions such as asthma. Negative cases were verified through self-reported negative COVID-19 tests. This wide-ranging control group strengthens the dataset's capacity to differentiate COVID-specific audio biomarkers. Positive cases were defined by reported COVID-19 tests and typical symptoms like fever, dry cough, or breathing difficulty. The clear labeling supported the training of classification models that can distinguish between COVID-19 and non-COVID-19 respiratory sounds.

Coswara Dataset [16] contains respiratory audio recordings and health metadata from 2,635 participants. The data were collected between April 2020 and February 2022. It spans data of non-COVID-19, COVID-19 positive, and recovered individuals from varied regions and age groups. The non COVID group included healthy individuals and those with other respiratory ailments such as asthma or colds, increasing model robustness. COVID-19 positive participants were classified into asymptomatic, mild, and moderate symptom categories, offering nuanced insights into symptom severity and their acoustic signatures. For consistency purposes, we considered only heavy cough samples from COVID-19 positive and non-COVID-19 groups.

The COUGHVID dataset [19] includes more than 25,000 cough recordings collected worldwide between April and December 2020. Each submission includes metadata on age, gender, respiratory conditions, and COVID-19 status. Over 2,800 recordings were annotated by physicians, who identified cough types (wet/dry), associated symptoms, and likely diagnoses (e.g., COVID-19, asthma). The dataset also features automated cough probability scores and self-reported information, supporting comprehensive analysis and labeling.

The AICovidVN dataset [20] contains cough recordings from individuals in Vietnam, classified into COVID-19 positive and COVID-19 negative classes.

UK COVID-19 Vocal Audio dataset [21] includes cough samples collected through the "Speak up and help beat coronavirus" survey. Participants were recruited via NHS Test and

Trace and the REACT-1 study. Each audio sample is linked to PCR test results, ensuring high reliability. Table I shows details of positive and negative COVID-19 cases in all six datasets discussed in this section.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

TABLE I: Details of Six COVID-19 Datasets

Dataset	Positive Samples	Negative Samples
CSS	158	567
IATos	2078	1493
Coswara (Heavy)	19	151
AlcovidVN	462	737
COUGHVID	547	547
UK COVID-19	645	650

B. Data Preprocessing

The acquired datasets were divided into two subsets. One subset was used for sound separation and the other for classification. The sound separation model was used to extract cough sounds from noisy and overlapping sound recordings and produce clean and isolated cough segments for further analysis. To maintain the integrity of the experimental design and prevent overfitting and data leakage, all cough recordings were excluded from the training, validation, and testing phase of the separation model. By separating both subsets, we ensured methodological consistency, bias reduction, improved robustness, and generalizability of the classification model in real world scenarios. The subset developed for sound separation was meticulously designed to support the creation of a robust deep learning model capable of detecting and isolating cough sounds within a complex acoustic environment. It comprised two primary components, that is, input mixtures and ground truth references.

The input mixtures were generated by combining isolated cough sounds with various noises, including conversations and ambient sounds typical of indoor offices and classrooms. These noises were sourced from Pixabay [22], Soundsnap [23] [24], and the LibriSpeech ASR corpus provided by OpenSLR [25] [26], ensures a wide and realistic range of acoustic environments. Pixabay provided ambient sound recordings, such as student activities and classroom chatter. However, Soundsnap provided even more specialized sound recordings, such as teacher instructions and varied classroom atmospheres that include quiet study areas and bustling lecture halls. The LibriSpeech ASR corpus offered low-noise, high-quality English speech recordings, suitable for clean speech conditions. This diversity allowed the dataset to simulate both straightforward and challenging acoustic scenarios. The ground truth references contained clean isolated cough sounds, including positive and negative COVID-19 samples, systematically organized for integration with background sounds.

The scarcity of large and well-annotated datasets is a significant challenge in sound separation research. No existing dataset provides large-scale cough sound mixtures combined with indoor background noise, making model training and evaluation difficult. Traditional datasets, based on real-world recordings, offer limited control over variables such as event timing, overlap, and signal-to-noise ratio (SNR), restricting their utility in developing robust separation models. Moreover, conventional data augmentation methods typically alter the soundscape as a whole without modifying individual sound events.

To address these limitations, we employed Scaper [27], an open-source Python library designed for procedural soundscape synthesis and augmentation. It enabled the generation of a controlled and

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

varied dataset by using a soundbank of isolated coughs and noises, functioning as a probabilistically controlled audio sequencer. Furthermore, it allowed precise control over parameters such as the number, timing, duration, overlap, and SNR of cough events. Using Scaper, we generated a standardized, diverse, and reproducible dataset tailored for training and evaluating cough sound separation models. The breakdown of the dataset is given below:

- Training Set: 12,000 mixtures created from 3,519 cough recordings and 729 noise samples.
- Validation Set: 1,500 mixtures created from 400 cough recordings and 913 noise samples.
- Test Set: 1,000 mixtures created from 492 cough recordings and 716 noise samples.

Each 5 -second mixture was sampled at 44.1 kHz , a standard for high-quality audio, and constructed using event specifications defining properties such as event type, timing, and SNR. The target class was "Cough". SNR values ranged uniformly from -5 dB to +5 dB , enabling a spectrum from subtle to prominent cough sounds. We applied data augmentation at the event level. Pitch shifting was drawn from a uniform distribution between -2 and +2 semitones, and time stretching varied from 0.9x to 1.1x speed. Each transformation had a 50% probability of being applied, ensuring varied and naturalistic sound variations. Furthermore, cough events were randomly positioned to prevent overfitting to specific temporal patterns.

The noise loudness was randomized using the reference loudness level of the background noise parameter in Scaper. It determines how loud the background audio will be about the sound events. This parameter is crucial in controlling the overall mix balance and ensuring variability in data.

The dataset used for classification was generated employing a technique aligned to the sound separation task while using unique cough sounds to promote diversity and prevent data leakage. This dataset created a realistic classroom environment by integrating cough sounds with other complex noises. This combination was processed using our trained sound separation model, which produced clean and noise-free cough segments. To support model generalization and maintain data integrity, new cough and noise samples were included for training the sound separation model. The final dataset consisted of 1,886 COVID-19-positive and 1,757 COVID-19-negative samples. Figure 2 illustrates our methodology for data preprocessing.

Separation of audio source signals from a combined single-channel signal involves the approximation of individual sources from their combined mixture [15]. Mathematically, expressed as:

$$x(t) = \sum_{i=1}^{s} y_i(t) \tag{2}$$

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

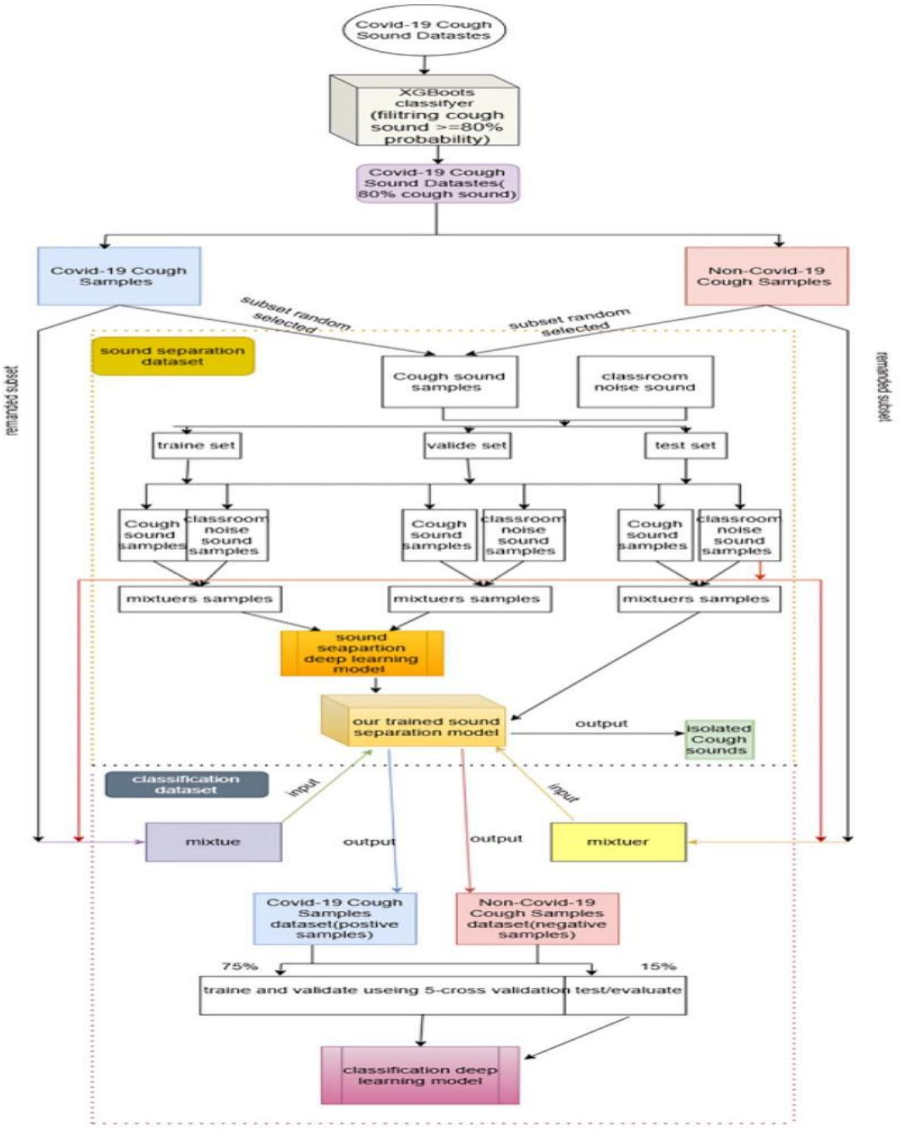


Fig. 2: Proposed method for data Preprocessing

Here, $x(t)$ and $y_i(t)$ denote the combined observed signal and the individual source signals. The total sources are denoted by S, implying that the mixture combines all the sources. To simplify the problem, we assume that the mixture contains only two different sources [15], denoted $s_1(t)$ and $s_2(t)$, leading to:

$$x(t) = s_1(t) + s_2(t) \tag{3}$$

This assumption mitigates the complexity of the separation task and makes it more computationally friendly. To proceed with further analysis and process the combined signal, the Short-Time Fourier Transform (STFT) is applied to convert the time-domain signal into a time frequency (TF). This transformation differentiates overlapping sounds by decomposing the signal into its frequency components [15]. The STFT representation of the mixed signal is given by:

$$X(n, f) = S_1(n, f) + S_2(n, f) \tag{4}$$

Here $X(n, f)$ denotes the STFT of the mixed signal, while $S_1(n, f)$ and $S_2(n, f)$ are the STFT representations of the original source signals. Here, n denotes the frame index (time segments), and f denotes the frequency index [15]. In source separation, the STFT representations of the sources remain

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

unknown and must be estimated from the combined observed signal. Here, we assume that only the magnitude spectrogram of the STFT is available, ignoring the phase information. This simplification facilitates the reconstruction of time-domain waveforms. The magnitude spectrogram of the measured audio signal is approximated [15] as:

$$|X_n| \approx |S_1(n,f)| + |S_2(n,f)| \tag{5}$$

Where $|X_n|$ represents the magnitude spectrogram of the mixed signal, and $|S_1(n,f)|$ and $|S_2(n,f)|$ denotes the magnitude spectrograms of the source signals. This approximation assumes that the observed magnitude spectra are a simple summation of the sources' magnitude spectra while ignoring phase information.

To approximate the unknown spectrograms $S_1(n,f)$ and $S_2(n,f)$ from the observed mixed spectrogram $X(n,f)$, deep neural networks (DNNs) are used to predict a time-frequency mask for each source. The masks are then applied to the observed mixture by multiplication by elements, effectively separating the individual STFT magnitude components corresponding to each speaker or sound source [15]. Once the magnitude STFT of a source is estimated, the Inverse ShortTime Fourier Transform (ISTFT) is employed to reconstruct the time-domain waveform. However, since phase information is not directly estimated, the resynthesized signal can contain noisy phase components of the original mixture. The STFT of a signal is computed as [15]:

$$X(t,f) = \sum_{n=1}^{N} x[n+tL]w[n]\exp\left(\frac{-j2\pi nf}{N}\right) \tag{6}$$

Here $x[n+tL]$ denotes the input signal in different time frames, $w[n]$ is a window function that ensures smooth segmentation, and the exponential term corresponds to the Fourier basis function, transforming the signal into the frequency domain. After acquiring the estimated spectrograms, ISTFT is applied to reconstruct the time-domain signals, is mathematically defined as [15]:

$$\hat{S}(n,f) = \frac{1}{N}\sum_{f=1}^{N} S_S(t,f)\exp\left(\frac{j2\pi nf}{N}\right) \tag{7}$$

Where $S_s(t,f)$ represents the estimated spectrogram of a source, and the exponential term acts as the inverse Fourier basis function, converting the representation of the frequency domain to the time domain [15].

The preprocessing step applies the Short-Time Fourier Transform (STFT) with a window length and a hop length of 512 and 128 respectively. This facilitates the extraction of spectral features from the audio signal. The neural network works in 257 frequency bins, representing the helpful portion of the spectrogram, using a sigmoid activation function to predict the mask. The model architecture includes two LSTM layers, each with 50 hidden units. The network predicts time frequency masks that are used for the mixed spectrogram to detach individual sources. It develops a single audio channel, suitable for monophonic audio. The Adam optimizer, which contains a learning rate of 1e − 3, is used to update the weights, while the L1 loss function finds the absolute difference between the actual and predicted magnitude sources. The model takes 50 training epochs, and both the training and processes validation datasets in mini-batches of 32 samples. Given the training data, the mixed signal magnitude spectrogram, X, is used as the input to the model, while the clean spectrograms of individual sources, S are used as the target output. The network's goal is to learn a mask that accurately identifies the proportion of each time-frequency bin in X_{tr} that corresponds to a specific source.

C. Sound Separation model

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

We developed a deep learning model to isolate a cough sound mixed with other sounds using time-frequency masking. We used pruning to optimize the model by improving its computational efficiency while maintaining performance. The model follows a sequential approach, where the input magnitude spectrogram of the mixture is changed into a log-scaled decibel (dB) representation stabilizing training by reducing the dynamic range of the spectrogram values.

The changed spectrogram is given to a Batch Normalization layer, which normalizes the feature distribution, preventing internal covariate shifts and speeding up the convergence during training. Then, Recurrent Stack is used to acquire temporal dependencies in the audio signal. These layers processing work on sequential spectrogram frames and capture long-term dependencies in the audio, and given an input sequence X_t, the LSTM updates its hidden state as defined in the equation:

$$f_t = \sigma\left(W_f x_t + R_f h_{t-1} + b_f\right)$$
$$i_t = \sigma(W_i x_t + R_i h_{t-1} + b_i)$$
$$o_t = \sigma(W_o x_t + R_o h_{t-1} + b_o)$$
$$c_t = c_{t-1} \odot f_t + j_t \odot i_t$$
$$h_t = \tanh(c_t) \odot o_t$$

Here, f_t, i_t, o_t denote the forget, input and output gates, respectively, while c_t is the cell state. The matrices W, R are the trainable weight parameters, b denotes the bias vector, and σ is the sigmoid activation function [15]. A unidirectional design was used to ensure real-time processing, which allowed the model to process sequential data efficiently without the need for future context, making it suitable for real-time applications [28]. Furthermore, pruning was applied to the linear transformation in the embedding layer using SNIP, which significantly reduced the number of active parameters and further optimized computational efficiency. Following the recurrent stack, the output is mapped to a mask space using the embedding layer, which applies a linear transformation defined as:

$$E = W_c H + b_e \tag{9}$$

Here, H denotes the hidden representation of the recurrent neural network and E denotes the embedding used to generate the final mask. The activation function (default: sigmoid) ensures that the mask values remain between 0 and 1 , is defined as:

$$\hat{M} = \sigma(E) \tag{10}$$

Finally, the predicted mask applied to the original magnitude spectrogram to estimate the separated source defined as:

$$\hat{S} = M \circ X \tag{11}$$

Here, M denotes the predicted mask, X denotes the input spectrogram, and $\hat{S}$ represents the estimated source. But when applying SNIP with pruning level equal to 0.01 in the linear layer, the function will be defined as:

$$E = W_e H + b_e \tag{12}$$

Here, H denotes the hidden representation of the recurrent neural network (RNN), and E denotes the embedding space for mask generation, which contains only 1% of its original weights, as 99% of the parameters in W_e have been pruned. The mathematical and computational consequences of this pruning are given below:

- Firstly, the weight matrix W_e contains zeros, which significantly reduces the memory footprint.
- Secondly, the model does not gain efficiency unless specialized sparse matrix computations are used. This is because dense operations still treat zero values as active parameters.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- Finally, substantial parameter reduction acts as an implicit regularizer, which mitigates overfitting and enhances generalization. Furthermore, because of pruning, a potential accuracy trade-off arises. With only 1% of the integrated parameters retained, the ability to generate accurate masks may be impacted. Since the mask is computed using the equation:

$$\hat{M} = \sigma(E) \tag{13}$$

Pruning affects the variability of E, which can reduce the generated mask structure and influence the final separation quality. The pruned mask is then applied to the original magnitude spectrogram to approximate the separated source, defined as:

$$\hat{S} = M \odot X \tag{14}$$

Here, M denotes the predicted mask. X denotes the input spectrogram, and $\hat{S}$ denotes the estimated separated source.

D. Classification Models

This study implements and evaluates the performance of the MobileNetV2, MobileNetV3-Small, and EfficientNet-Lite-0 architectures for real-time cough-based classification of COVID-19. These architectures were chosen for their lightweight design, low computational requirements, and efficiency in real-time settings. Such models are suitable for scenarios where memory, processing power, and energy are constrained and high real-time accuracy is the requirement. All models were trained using the same dataset, preprocessing pipeline, and hyperparameters to ensure fair evaluation.

MobileNetV2 [29] is an advanced iteration of MobileNet that introduced a depthwise separable convolutions technique that significantly reduces computational cost, specifically designed for mobile and embedded applications where computational efficiency is crucial. It proposed an inverted residual and linear bottleneck layer, which improves accuracy and performance while maintaining a lightweight structure. It contains three primary components, i.e., standard convolutional layers, inverted residual blocks, and fully connected layers. The inverted Residual Block is the core component of the proposed architecture. The model first grows the input channels before using depthwise convolutions, making sure that most computations occur in a high-dimensional space while maintaining a low-dimensional bottleneck at the end. The depthwise convolution significantly minimizes computation by processing each input channel in a different way instead of applying a full convolution across all channels [30]. The application of skip connections is another key characteristic of the inverted residual block. It supports information flow through the network without modification. ReLU6 is a variant of ReLU that caps values at 6 to improve robustness against quantization errors. The first layer of the network is a 3×3 standard convolution with a stride of 2 , which increases the receptive field while downsampling the input. This is followed by a sequence of inverted residual blocks. The network progressively improved the depth of the channel while mitigating spatial dimensions, making it efficient in both computational cost and memory footprint. The final stage of MobileNetV2 contains a convolutional layer 1×1, a global average pooling layer, and a fully connected classifier. The global average pooling layer mitigates the spatial dimensions to 1×1, effectively transforming feature maps into a compact feature vector, followed by a dropout layer that minimizes overfitting before the final classification. MobileNetV2 achieves a remarkable balance between accuracy, computational efficiency, and memory footprint that makes it an ideal architecture for realworld applications requiring energy efficiency and quick inference. MobileNetV3 is an improved version of MobileNetV2, proposed by [31] using network architecture search (NAS). This technique was used to search for the best kernel size and find the optimized MobileNet architecture to fulfill the low resourced hardware platforms in terms of size, performance, and latency. The MobileNetV3 introduced building blocks inspired by the previous versions [32].

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

A significant innovation in MobileNetV3 is the introduction of the **h**-swish activation function, an efficient approximation of the swish function. The original swish function is defined as swish$(x) = x \cdot \sigma(x)$, where $\sigma(x)$ is the sigmoid function. While swish has been shown to improve performance over

ReLU, its computational cost makes it less suitable for mobile applications. The h -swish function, on the other hand, is defined as h-swish $(x) = x \cdot \frac{\text{ReLU6}(x+3)}{6}$, where ReLU 6 $(x) = \min(\max(0, x), 6)$. This formulation eliminates the need for the expensive exponential operations inherent in the sigmoid function, making h-swish significantly more efficient while retaining similar performance benefits.

The MobileNetV3 block contains the inverted residual block, which includes a depthwise separable convolution block and a squeeze-and-excitation block [33]. The inverted residual block is inspired by the bottleneck blocks [34], where it uses an inverted residual connection to connect the input and output features on the same channels and improves the feature representations with low memory usage. The depthwise separable convolutional contains a depthwise convolutional kernel applied to each channel and a 1×1 pointwise convolutional kernel with batch normalization layer (BN) and the ReLU or h -swishh-swish activation functions. The squeeze-and-excitation (SE) block is used to pay more attention to the relevant features on each channel during training.

MobileNetV3-Small is the best suitable model for low latency and resource-constrained settings. It contains three main sections, i.e., the stem, the bottleneck blocks, and the head. The stem contains an initial 3×3 convolution layer with stride 2 , followed by hard-swish activation to acquire low-level image features while mitigating spatial dimensions. This early downsampling is critical, as it helps minimize computational costs in later stages. The bottleneck blocks, which form the network's core, utilize inverted residuals with linear bottlenecks. Each bottleneck block consists of an expansion layer, depthwise convolution for spatial filtering, an optional SE block for channel-wise feature recalibration, and a projection layer to reduce feature depth. These blocks are carefully designed to balance efficiency and feature extraction, using ReLU activation in early layers for speed and switching to hard-swish in later layers for improved non-linearity. A small variant of MobileNetV3 employs 5×5 depthwise convolutions more frequently, as these have been found to capture spatial dependencies more effectively while maintaining a low computational footprint. Its integrated SE blocks in multiple bottleneck layers, allow the network to adjust the importance of different channels dynamically. This significantly improves feature selection without adding excessive complexity. Furthermore, hard-swish activation replaces traditional ReLU in critical layers, further optimizing the network for speed and accuracy. The final head section of the network comprises a 1×1 convolution to increase feature depth, followed by another SE block, Global Average Pooling (GAP), and a fully connected (FC) layer with 1024 neurons before passing through a dropout layer to prevent overfitting. The final layer is a softmax classifier that outputs predictions based on the number of target classes.

EfficientNet-Lite-0 [35] [36] is a latest lightweight neural network architecture, designed to improve accuracy and computational efficiency, specifically for mobile and resource constrained environments. It is an extension of the MobileNetV2 framework and utilizes inverted bottleneck convolutional layers (MBConv blocks) as its core components. These MBConv blocks leverage depthwise separable convolutions and linear bottlenecks. The compound scaling approach is the key innovation of EfficientNet-Lite-0, which improves the performance of the model by uniformly scaling three fundamental dimensions, i.e., depth, width, and input resolution. Unlike conventional scaling strategies that adjust only one dimension at a time, compound scaling balances all three dimensions, which improves efficiency and accuracy.

EfficientNet employs compound scaling based on the principle that increasing image resolution requires proportionally deeper and wider networks to effectively capture fine-grained details. Empirical grid search has determined the optimal scaling coefficients for EfficientNet-Lite-0.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

These coefficients ensure efficient model scaling without increasing the parameter overhead. The EfficientNet-Lite-0 architecture composed of standard convolutional layers, MBConv layers with varying kernel sizes ($k = 3 \times 3$ and $k = 5 \times 5$), a final 1×1 convolution, global pooling, and fully connected layers. The model architecture ensures an optimal balance between accuracy and computational cost.

E. Feature Extraction

This study employed Mel-spectrogram-based feature extraction to capture the key characteristics of cough sounds, leveraging its perceptual alignment with human hearing for effective analysis. Audio recordings were resampled to 22,050 Hz, and Mel spectrograms were computed using 128 Mel bands with a hop length of 512, and an FFT window size of 2048. Recordings were standardized to 4 seconds in duration. Log power scaling (dB) and feature normalization were applied to enhance interpretability and reduce variability from recording conditions.

The extracted spectrograms work as a single-channel input to the selected model for the binary classification of cough sounds. The models were trained using 5 -fold stratified cross validation to ensure robustness and handle class imbalance. Cross-entropy loss was used as the objective function. The dataset was divided into training, validation, and test sets, with 15% held for the final evaluation. During each fold, model performance was tracked via training / validation loss and accuracy, and the best checkpoint was used for testing. Training and evaluation of the model was conducted using Paperspace, a cloud-based platform that provides high-performance GPU resources. A Free-A6000 instance with 8 CPU cores, 45 GB of RAM, and 48 GB of GPU memory was used, enabling efficient handling of largescale deep learning tasks and iterative experimentation without local hardware constraints.

F. Sound Separation Models

This section presents a detailed evaluation of cough sound isolation using the original and pruned mask-based separation models, tested on a dataset comprising 1,000 mixtures. Five audio quality metrics-SDR, ISR, SIR, SAR, and SNR were used in conjunction with computational metrics such as model size, throughput, FLOPs representing the total number of arithmetic operations, Multiply-Accumulate Operations (MACs) assign equal weight to multiplications and additions, in contrast to FLOPs [37]. Typically, FLOPs $\approx 2 \times$ MACs to assess computational load. Furthermore, throughput and latency were employed to evaluate inference speed, responsiveness, and the number of trainable parameters reflecting model complexity to assess the efficiency and resource consumption of the models [38].

- SDR evaluates total distortion-unwanted changes in the waveform of the audio and is defined as:

$$\mathrm{SDR} = 10\log_{10} \frac{\|s\|^2}{\left\|e_{\mathrm{spat}} + e_{\mathrm{interf}} + e_{\mathrm{artifif}}\right\|^2} \tag{15}$$

- SAR measures artificing-unintended sounds introduced through digital processing and is defined as:

$$\mathrm{SAR} = 10\log_{10} \frac{\left\|s + e_{\mathrm{spat}} + e_{\mathrm{interf}}\right\|^2}{\|e_{\mathrm{artifif}}\|^2} \tag{16}$$

- SIR quantifies interference-noise from other unwanted sources in the extracted signal and is defined as:

$$\mathrm{SIR} = 10\log_{10} \frac{\left\|s + e_{\mathrm{spat}}\right\|^2}{\|e_{\mathrm{interf}}\|^2} \tag{17}$$

- ISR assesses spatial accuracy-how closely the perceived sound location matches the original and is defined as:

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

$$\text{ISR} = 10\log_{10} \frac{\|s\|^2}{\|e_{\text{spat}}\|^2} \tag{18}$$

- SNR quantifies the level of sensor noise relative to the target signal and interference and is defined as:

$$\text{SNR} = 10\log_{10} \frac{\|s + e_{\text{interf}}\|^2}{\|e_{\text{noise}}\|^2} \tag{19}$$

These metrics, expressed in decibels (dB), quantify various forms of distortion in the extracted signal. The clean source signal s is approximated by $\hat{s} = s + e_{\text{spat}} + e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}}$, where each error term corresponds to a specific distortion type.

G. Performance Evaluation of Models

The performance of deep learning models was evaluated using confusion matrix, which is composed of True Positives, False Positives, True Negatives, and False Negatives. From which key metrics were derived: accuracy (overall correctness), precision (true positives over predicted positives), recall (true positives over actual positives), specificity (true negatives over actual negatives), and F1-score (harmonic mean of precision and recall)

IV. Results and Discussion

It is evident from the results of the test dataset of 1,000 mixtures that both the original and pruned models significantly improved cough sound isolation over the raw mixtures. This study highlights the effectiveness of the proposed cough separation method in improving the quality of isolated cough from mixed audio tracks. Before applying the separation method, the SDR was -2.35, indicating a severe distortion in the cough sounds. After applying the model, the SDR improved to 5.99 , confirming that the separation process significantly decreased distortions and improved the quality of the extracted cough sounds. Similarly, the SIR increased from -1.31 to 9.29, highlighting the model's ability to isolate the cough components from other background noises. The SNR improved from -2.11 to 6.54, demonstrating a substantial reduction in unwanted noise. However, the SAR dropped from 189.25 to 3.93, indicating the introduction of some artifacts in the process. The ISR also decreased from 21.02 to 12.19 , suggesting some loss of spatial information. However, a reduction in SAR and ISR may indicate minor trade-offs in separation quality. Significant improvements in SDR, SIR, and SNR confirm that the model successfully extracts clear cough sounds, making them suitable for cough classification. Pruning was used to improve the computational efficiency of the cough separation model. The pruned model was evaluated using the same metrics to determine its affect on separation quality. The results showed a slight decrease in SDR, dropping from 5.99 to 5.14, indicating a slight increase in distortion. However, SIR improved from 9.29 to 9.80 , suggesting that pruning slightly improved cough isolation from background sounds. The SNR decreased from 6.54 to 5.52, indicating a slight increase in background noise, while the SAR dropped from 3.93 to 2.73 , showing an increase in artifacts. The ISR decreased from 12.19 to 6.33, suggesting a loss of spatial information. Despite these minor trade-offs, the pruned model still performs well in cough separation, maintaining high clarity levels and interference reduction. The improvement in SIR suggests that pruning may have helped focus the model's processing power on isolating cough more effectively .

TABLE II: Performance comparison between original and pruned sound separation models

Metric	Without separation	Original sound separation	Pruned separation
SDR	-2.35	5.99	5.14
SAR	189.25	3.93	2.73
SIR	-1.31	9.29	9.80

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

ISR	21.02	12.19	6.33
SNR	-2.11	6.54	5.52

In addition to its impact on separation quality, pruning significantly improved the computational efficiency of the model. The number of parameters was reduced from 95.921 K to 82.714 K , leading to a 13.76% decrease in model size (from 0.37 MB to 0.32 MB). This reduction also resulted in (FLOPs), which dropped from 334.035 M to 289.729 M , and (MACs), which decreased from 167.018 M to 144.864 M . The pruning process also optimized the model's inference time on CPU, reducing it from 0.151 seconds to 0.057 seconds, representing a 62.25% improvement in processing speed. Furthermore, throughput increased from 7.26 inferences per second to 8.39 inferences per second, reflecting a 15.56% boost in efficiency. These improvements indicate that the pruned model can process more audio files in less time, making it a more suitable choice for real-time applications or deployment on resource constrained devices .

SDR, SIR, and SNR are considered the most relevant metrics when applying the cough separation model as a preprocessing step for cough classification. High SDR ensures that the extracted cough has minimum distortion, which ensures classification accuracy. A high SIR confirms that the cough sound is well isolated from other noise interference, allowing classification models to focus solely on the characteristics of the cough. A high SNR ensures that the extracted cough sounds are free from excessive noise, making them suitable for classification.

After applying pruning, the model maintained strong separation performance with only slight trade-offs in SDR and SNR while improving SIR. A decrease in SAR and ISR suggests a small increase in artifacts and a loss of spatial information, but these are less critical for classification tasks. More importantly, pruning significantly improved the efficiency of the model, reducing inference time by 62.25% and increasing throughput by 15.56%. The evaluation demonstrates that both the original and pruned models substantially improved the cough sound isolation performance compared to the mixture. The original model excels in terms of isolation quality, whereas the pruned model offers significant computational savings. The pruning offers a good balance between efficiency and performance, making it a practical choice for applications requiring fast and accurate cough separation.

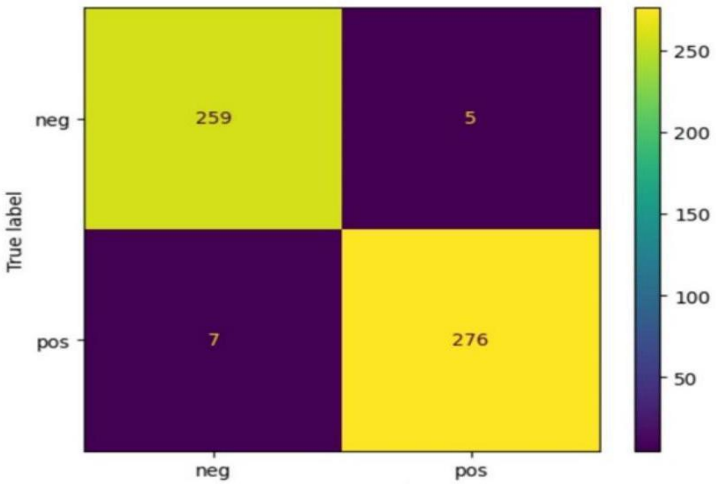


Fig. 3: Confusion Matrix for MobileNetV2

MobileNetV2 achieved an accuracy of 97.8%, a precision of 97.5%, recall of 97.5%, and F1-score of 97.8%. It correctly classified 259 negative cases and 276 positive cases, whereas, misclassified 5 negative cases as positive and 7 positive cases as negative.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

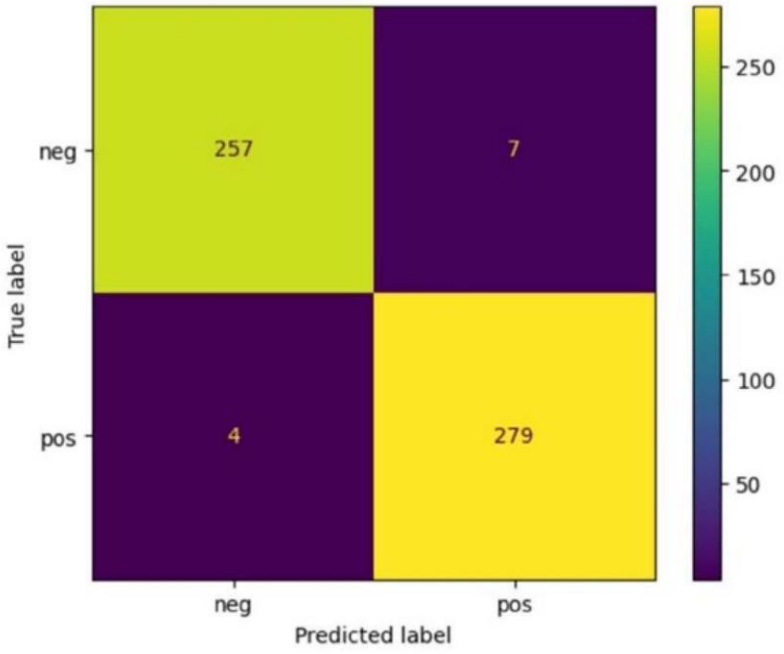
Research Article

EfficientNet-Lite-0 achieved an accuracy of 97.9%, with a precision of 97.5%, recall of 98.5%, and F1-score of 98.0% . The confusion matrix shows that it correctly classified 257 negative cases and 279 positive cases. Whereas, it misclassified 7 negative cases as positive and 4 positive cases as negative.

MobileNetV3-Small outperformed both previous models in terms of classification performance, achieving the highest accuracy of 99.0% and a precision of 98.9%, recall of 99.2%,

TABLE III: Model Comparison based on Complexity

Model	No. Params	Model Size	FLOPs	MACs	Latency (s)	Throughput (CPU)	Throughput (CUDA)
MobileNetV2	2,225,858	8.49 MB	292.642 M	146.321 M	0.0086 sec on (CPU), 0.0033 sec on (CUDA)	57.63inf/s on CPU	323.26inf/s on CUDA
MobileNetV3 Small	1,519,618	5.80 MB	54.499 M	27.25 M	0.0060 sec on (CPU), 0.0041 9 sec on (CUDA)	99.94inf/s on CPU	245.25inf/s on CUDA
EfficientNet-lite-0	4,651,432	17.74 MB	375.117 M	187.558 M	0.0083 sec on (CPU), 0.0030 9 sec on (CUDA)	73.58inf/s on CPU	316.40inf/s on CUDA



Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Fig. 4: Confusion matrix for EfficientNet-lite-0.

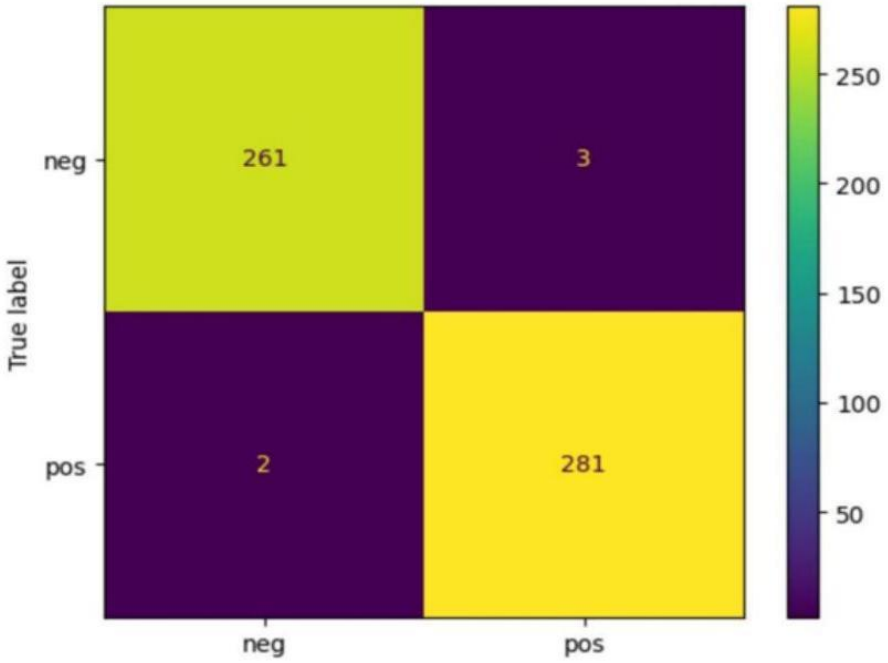


Fig. 5: Confusion Matrix for MobileNetV3-Small

and F1-score of 99.1%. The confusion matrix it correctly classified 261 negative cases and 281 positive cases. Whereas, it misclassified only 3 negative cases as positive and 2 positive cases as negative, indicating fewer misclassifications than other models.

In terms of efficiency, EfficientNet-Lite-0 had 4,651,432 parameters and a model size of 17.74 MB , with a FLOPs count of 375.11 M and MACs of 187.558 M . Its inference time was 0.0083 seconds on a CPU, achieving a throughput of 73.58 inferences per second. MobileNetV2 had fewer parameters 2,225,858 and a smaller model size of 8.49 MB , which

TABLE IV: Performance Evaluation of deep learning model

Metric	MobileNetV2	MobileNetV3-Small	EfficientNet-Lite0
Accuracy	97.8	99.0	97.9
Precision	97.5	98.9	97.5
Recall	97.5	99.2	98.5
F1-score	97.8	99.1	98.0

reduced its computational complexity. The FLOPs count was 292.642 M, and MACs were 146.321M. However, its inference time was slightly slower at 0.0086 seconds, resulting in a lower throughput of 57.63 inferences per second. Furthermore, MobileNetV3-Small was the most computationally efficient model, with only 1,519,618 parameters and a significantly smaller model size of 5.80 MB . The FLOPs count was reduced to 54.499 M , and MACs were only 27.250 M , making it much less computationally

intensive. It also had the fastest inference time of o. 0061 seconds, achieving the highest throughput at 99.94 inferences per second .

Results revealed that MobileNetV3-Small is most optimal in terms of both classification performance and computational efficiency. It achieved the highest accuracy, precision, recall, and F1 score while requiring the least computational resources. The other two models had almost similar performance, but the first model had a higher throughput than the second, making it better suited for real-time applications. The second model was more lightweight but had slightly lower throughput and inference speed. Overall, MobileNetV3-Small is the best choice for real-time detection and classification of cough-based COVID-19.

V. Conclusion and Future Work

This study proposed a Lightweight deep learning approach for real-time cough detection and COVID-19 classification. Six datasets i.e., CSS, AITos, Coswara, AlcovidVN, COUGHVID, and UK COVID-19 were acquired and the proposed lightweight sound separation deep learning model based on LSTM was used to eliminate unwanted noises from cough sounds. After that, Signal-to-Distortion Ratio (SDR), Signal-to-Artifacts Ratio (SAR), Signal-to-Interference Ratio (SIR), Image-to-Spatial Ratio (ISR), and Signal-to-Noise Ratio (SNR) were used to assess the quality of the cleaned cough sounds. MobileNetV2, MobileNetV3 Small, and EfficientNet-lite-0 were implemented to classify cough sounds as COVID-19 positive and COVID-19 negative. MobileNetV2, MobileNetV3 Small, and EfficientNet-lite-0 achieved an accuracy of 97.8%, 99%, and 97.9% respectively. MobileNetV3 Small was proved to be the best model in terms of classification accuracy and computational efficiency. In the future, our goal is to improve pruning techniques to mitigate their impact on separation quality while improving computational efficiency. Furthermore, we plan to explore more advanced separation techniques or hybrid models to improve the trade-off between quality and efficiency.

References

[1] S. Jin, G. Liu, and Q. Bai, "Deep learning in COVID-19 diagnosis, prognosis and treatment selection," Mathematics, vol. 11, no. 6, p. 1279, 2023.

[2] World Health Organization. (2025) Covid-19 dashboard - deaths. Accessed: 2025-04-04. [Online]. Available: https://data.who.int/dashboards/covid19/deaths?m49=001

[3] L. Cheng, L. Lan, M. Ramalingam, J. He, Y. Yang, M. Gao, and Z. Shi, "A review of current effective COVID-19 testing methods and quality control," Arch. Microbiol., vol. 205, no. 6, p. 239, 2023.

[4] R. Mardani, A. Ahmadi Vasmehjani, F. Zali, A. Gholami, S. D. Mousavi Nasab, H. Kaghazian, M. Kaviani, and N. Ahmadi, "Laboratory parameters in detection of COVID-19 patients with positive RTPCR; a diagnostic accuracy study," Arch. Acad. Emerg. Med., vol. 8, no. 1, p. e43, 2020.

[5] Y. Shi, H. Liu, Y. Wang, M. Cai, and W. Xu, "Theory and application of audio-based assessment of cough," J. Sens., vol. 2018, pp. 1-10, 2018.
[6] T. K. Dash, S. Mishra, G. Panda, and S. C. Satapathy, "Detection of COVID-19 from speech signal using bio-inspired based cepstral features," Pattern Recognit., vol. 117, no. 107999, p. 107999, 2021.

[7] K. K. Lella and A. Pja, "Automatic diagnosis of COVID-19 disease using deep convolutional neural network with multi-feature channel from respiratory sound data: Cough, voice, and breath," Alex. Eng. J., vol. 61, no. 2, pp. 1319-1334, 2022.

[8] A. Imran, I. Posokhova, H. N. Qureshi, U. Masood, M. S. Riaz, K. Ali, C. N. John, M. I. Hussain, and M. Nabeel, "AI4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app," Inform. Med. Unlocked, vol. 20, no. 100378, p. 100378, 2020.

[9] M. Soltanian and K. Borna, "Covid-19 recognition from cough sounds using lightweight separable-quadratic convolutional network," Biomed. Signal Process. Control, vol. 72, no. 103333, p. 103333,

2022.
[10] G. Chaudhari, X. Jiang, A. Fakhry, A. Han, J. Xiao, S. Shen, and A. Khanzada, "Virufy: Global applicability of crowdsourced and clinical datasets for AI detection of COVID-19 from cough," 2020.

[11] M. Loey and S. Mirjalili, "COVID-19 cough sound symptoms classification from scalogram image representation using deep learning models," Comput. Biol. Med., vol. 139, no. 105020, p. 105020, 2021.

[12] N. Turpault, S. Wisdom, H. Erdogan, J. Hershey, R. Serizel, E. Fonseca, P. Seetharaman, and J. Salamon, "Improving sound event detection in domestic environments using sound separation," 2020.

[13] A. Musa, H. A. Kakudi, M. Hassan, M. Hamada, U. Umar, and M. L. Salisu, "Lightweight deep learning models for edge devices-a survey," Int. J. Comput. Inf. Syst. Ind. Manag. Appl., vol. 17, p. 18, 2025.
[14] N. Lee, T. Ajanthan, and P. H. S. Torr, "SNIP: Single-shot network pruning based on connection sensitivity," 2018.
[15] R. J. Issa and Y. F. Al-Irhaym, "Audio source separation using supervised deep neural network," J. Phys. Conf. Ser., vol. 1879, no. 2, p. 022077, 2021.

[16] D. Bhattacharya, N. K. Sharma, D. Dutta, S. R. Chetupalli, P. Mote, S. Ganapathy, C. Chandrakiran, S. Nori, K. K. Suhail, S. Gonuguntla, and M. Alagesan, "Coswara: A respiratory sounds and symptoms dataset for remote screening of SARS-CoV-2 infection," Sci. Data, vol. 10, no. 1, p. 397, 2023.

[17] D. T. Pizzo and S. Esteban, "IATos: AI-powered pre-screening tool for COVID-19 from cough audio samples," 2021.

[18] C. Brown, J. Chauhan, A. Grammenos, J. Han, A. Hasthanasombat, D. Spathis, T. Xia, P. Cicuta, and C. Mascolo, "Exploring automatic diagnosis of COVID-19 from crowdsourced respiratory sound data," in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York, NY, USA: ACM, 2020.

[19] L. Orlandic, T. Teijeiro, and D. Atienza, "The COUGHVID crowdsourcing dataset, a corpus for the study of large-scale cough analysis algorithms," Sci. Data, vol. 8, no. 1, p. 156, 2021.

[20] L. H. Nguyen, N. T. Pham, V. H. Do, L. T. Nguyen, T. T. Nguyen, H. Nguyen, N. D. Nguyen, T. T. Nguyen, S. D. Nguyen, A. Bhatti, and C. P. Lim, "Fruit-CoV: An efficient vision-based framework for speedy detection and diagnosis of SARS-CoV-2 infections through recorded cough sounds,"Expert Syst. Appl., vol. 213, no. 119212, p. 119212, 2023.

[21] J. Budd, K. Baker, E. Karoune, H. Coppock, S. Patel, R. Payne, A. Tendero Cañadas, A. Titcomb, D. Hurley, S. Egglestone, L. Butler, J. Mellor, G. Nicholson, I. Kiskin, V. Koutra, R. Jersakova, R. A. McKendry, P. Diggle, S. Richardson, B. W. Schuller, S. Gilmour, D. Pigoli, S. Roberts, J. Packham, T. Thornley, and C. Holmes, "A large-scale and PCR-referenced vocal audio dataset for COVID-19," Sci. Data, vol. 11, no. 1, p. 700, 2024.

[22] Pixabay, "Classroom sound effects," 2025, accessed: 2025-04-09. [Online]. Available: https://pixabay.com/soundeffects/search/classroom/

[23] Soundsnap, "Classroom sound effects," 2025, accessed: 2025-04-09. [Online]. Available: https://www.soundsnap.com/tags/classroom

[24] -, "Teacher sound effects," 2025, accessed: 2025-04-09. [Online]. Available: https://www.soundsnap.com/tags/teacher

[25] OpenSLR, "Librispeech asr corpus," 2025, accessed: 2025-04-09. [Online]. Available: https://www.openslr.org/12

[26] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2015.

[27] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in 2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA). IEEE, 2017.

[28] S. Venkatesh, A. Benilov, P. Coleman, and F. Roskam, "Real-time low-latency music source separation using hybrid spectrogram-TasNet," 2024.

[29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2018.

[30] K. Dong, C. Zhou, Y. Ruan, and Y. Li, "MobileNetV2 model for image classification," in 2020 2nd International Conference on Information Technology and Computer Application (ITCA). IEEE, 2020.

[31] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2019.

[32] M. Abd Elaziz, A. Dahou, N. A. Alsaleh, A. H. Elsheikh, A. I. Saba, and M. Ahmadein, "Boosting COVID-19 image classification using MobileNetV3 and aquila optimizer algorithm,"Entropy (Basel), vol. 23, no. 11, p. 1383, 2021.

[33] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "MnasNet: Platform-aware neural architecture search for mobile," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2019.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[35] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," 2019.

[36] P. Pintea, "Is machine learning energy-efficient enough to be used in mobile fashion applications?" Master's thesis, University of Groningen, 2022, master's thesis.

[37] E. Cano, D. FitzGerald, and K. Brandenburg, "Evaluation of quality of sound source separation algorithms: Human perception vs quantitative metrics," in 2016 24th European Signal Processing Conference (EUSIPCO). IEEE, 2016.

[38] H.-I. Liu, M. Galindo, H. Xie, L.-K. Wong, H.-H. Shuai, Y.-H. Li, and W.-H. Cheng, "Lightweight deep learning for resource-constrained environments: A survey," ACM Comput. Surv., vol. 56, no. 10, pp. 1-42, 2024.