ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Wiener Filter with Convolutional Neural Network for Noise Removal in API-Based AI Models

Joel Ryan A. de Guzman[1], Robert G. de Luna[2] and Marife A. Rosales[3]

ABSTRACT

This research aims to develop a robust Application Program Interface (API)-Based Artificial Intelligence (AI) system for effective noise removal from audio signals, enhancing speech quality and intelligibility in noisy environments to be fed into different AI models to assess the applicant interview. The proposed methodology combines sophisticated signal processing techniques and noise reduction algorithms with AI models trained on clean voice data and noise patterns. To achieve this goal, we leverage two key components: the Wiener filter and a Convolutional Neural Network (CNN). The Wiener filter serves as the foundational noise reduction technique, exploiting statistical properties of the signal and the noise to suppress unwanted noise components effectively. Concurrently, CNN is integrated to classify the clean and noisy audio. In this research, the best optimizers selected, including Adam, SGD, RMSprop, Adagrad, and Adadelta are evaluated to identify the most suitable classification. The optimizers evaluated through cross-validation and hold-out validation in the same batch size (25) and epoch (25) were used. The study demonstrates that the Adam optimizer yields the best results. The epoch was optimized to 35, 75, 105, and 125 and epoch of 105 was selected with accuracy of 99.52%, Recall of 100%, F1-Score of 99.50%, and ROC_AUC of 99.99% for cross-validation while Accuracy of 98.79%, Recall of 99.21%, F1-Score of 98.81%, and ROC_AUC of 99.54% for hold-out validation, significantly improving AI model performance. Lastly, we ensured the batch size parameter was suitable for our model by tuning it with different settings (25, 50, 75, and 125) using the optimized optimizer and epoch. The batch size of 25 yielded the best accuracy. The modeled CNN also included kernel regularization L2 to avoid overfitting.

Article information:

Keywords: API pipeline, Business process outsourcing, Convolutional Neural Network, Mel-Spectrogram, Wiener Filter

Article history:
Received: March 22, 2024
Revised: June 27, 2024
Accepted: July 11, 2024
Published: September 14, 2024 (Online)

DOI: 10.37936/ecti-cit.2024184.256162

1. INTRODUCTION

Business Process Outsourcing (BPO) is a subset of outsourcing in which the operations and responsibilities of a certain business process are contracted out to a third-party service provider. One example of the BPO industry is the "Call Center," where the mode of communication is through calls dealing with clients. According to the 2022 Outsourcing Performance Report of Outsource Accelerator (OA), a marketplace for the BPO industry, call centers in the Philippines ranked fifth. The country received a disproportionate share of outsourcing-related inquiries—3.81 percent of all inquiries indicating that international companies were considering locating their back offices in the Philippines. Hiring is continuous to recruit the best applicants so that the company can satisfy the job requirements, where skills must be needed to demonstrate their communication skills and attitude to be on top of the list. Call centers are very diligent in selecting their candidate employees to satisfy the needs of their clients. The hiring process traditionally takes months for the entire process before deploying the applicant and depends on the company's training program; this is because, after the interview process and applicant selections, the company will conduct training to develop the skills and assess again for the final judgment of hiring. The issue that this method revealed will cause the applicant's cost and hiring pro-

[1,2,3]The authors are with the Polytechnic University of the Philippines, E-mail: djoelryan@yahoo.com, rgdeluna@pup.edu.ph and marosales@pup.edu.ph

cesses to be delayed. Several companies in this industry embracing artificial intelligence, believing to address the issues and expenses. Thus, the potential of artificial intelligence significantly impacted the company in selecting the applicants without delaying the deployment, bias, and cost reduction. The whole process will change embarking on the new technology that AI brings. However, our research is focused on an API pipeline that will give a robust method to eliminate noise from audio. A major barrier to successful speech recognition and the best possible performance of AI models in audio data processing is noise. Different kinds of noise, like background noise and environmental factors, can deteriorate the quality of audio signals and cause mistakes in recognition systems when transcribing the audio to text. Thus, a critical aspect of this study involves using automated transcription systems to convert audio interviews into text for fair and consistent applicant grading. To assess the impact of noise, we transcribed audio samples without applying a Wiener filter, revealing significant errors. Researchers analyzed these samples to identify misspelled words and noise presence, laying the groundwork for developing robust noise mitigation methods to enhance transcription accuracy. This process included using a speech recognizer to convert speech to text and thoroughly reviewing the transcriptions for noise-induced errors. The suggested method employs a Convolutional Neural Network (CNN) model to accurately detect the presence of noise in audio samples and real audio data. The problem of applying AI in the proposed system is the integrity and quality of the audio from the applicant where the system should accurately clean and detect the presence of noise before transcription of audio to text. Hence the researcher introduced an innovative approach, and processing this problem in an API is suitable. Our research endeavors to devise a resilient innovation for efficaciously eliminating noise from audio signals, hence augmenting speech quality and comprehensibility. The main objectives are to optimize the noise detection model, develop an efficient Wiener filter for noise removal, and explore the integration of multi-model processing for enhanced speech analysis. The suggested approach uses noise reduction methods in concurrence with sophisticated signal processing techniques to recover audible speech from noisy audio settings. Wide-ranging effects may result from the effective use of this technology in several industries, including voice assistants, call centers, and audio analytics applications. By tackling noise interference in speech recognition, this study has the potential to greatly increase AI models' performance and reliability in noisy audio situations. This research introduces the detection of noise from audio using CNN and cleaning through a Wiener filter before transmitting into different systems that the audio will use for the assessments as shown in Figure 1. This research project aims to develop an innovative noise detection and removal methodology for improving speech quality in audio data processing.

2. REVIEW OF RELATED WORKS

The adoption of Artificial Intelligence nowadays is vastly looking for the opportunity to leverage its application. One of the industries is call centers wherein many researchers look for opportunities and change the traditional process of hiring applicants [1-2]. In this way, the cycle time of the hiring process will be lessened, and accurate without any bias during the assessment and cost reduction [3]. Various studies have been conducted on the automation of the recruiting process, including studies on emotion and

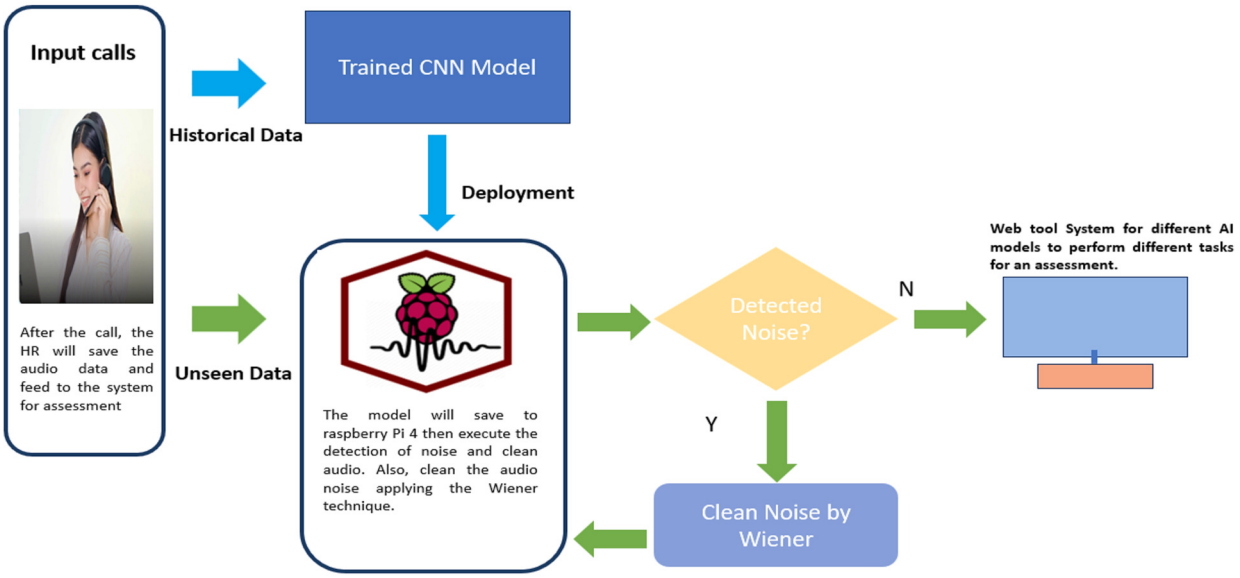


Fig.1: *The Proposal System API of Noise Cancellation.*

auditory cues where the pre-emphasis approach is employed, the size of certain frequencies, and how P (n) is amplified based on other frequencies to increase the total signal to noise ratio (SNR) [4] and using natural language processing and machine learning [5] in overcoming the noise in audio with audio API to address the issue of speech enhancement and transcribing text from audio is very challenging, there is various study on Audio API and suppressing the noise, and [6] discussed from their research the audio nodes (Audio nodes) allowing audio processing to follow a certain flow. API for Web Audio offers a resource for de-creating applications for interactive music and sound in Web Audio XML [7]. Microsoft Speech API and Google Cloud Speech API were introduced to address the challenge of speech [8-9]. AEC research has advanced significantly with the ICASSP 2022 Acoustic Echo Cancellation Challenge, which introduces mobile scenarios, a 48 kHz audio standard, and a novel speech recognition measure. The challenge encourages cooperation and assessment by providing open-source large datasets, an online subjective test framework, and an objective metric service. The average Mean Opinion Score and speech recognition of the word accuracy rate are used to decide the winners [10]. However, different research studies introduced noise reduction, speech enhancement, and Wiener Filtering noise. N. Alamdari et al. developed an application for noise reduction using an unsupervised noise classifier that used Wiener filtering for noise reduction and was specifically designed to automatically adjust to various ambient backgrounds and speaker configurations in N-HANS architecture [11]. Thus, the study used two equivalent neural networks composed of stacks of residual blocks, each of which is trained on extra speech- and noise-based recordings using auxiliary sub-networks [12]. The far-end user must be able to hear this noise and mute it without impairing the speaker's speech. Because non-stationary noise from the background eludes typical algorithms' detection and suppression, it negatively impacts the far-end user's experience with the quality of call audio. Through the implementation of real-time noise reduction in the sender-side browser, this study aims to increase portability [13]. Moreover, there are techniques for filtering the noise using a Wiener filter. Kumar et al. (2020) developed an effective Wiener filter and used it in conjunction with a digital hearing aid, a real-valued fast Fourier transform (FFT), and a real-valued inverse FFT processor to reduce noise [14]. One method for lowering background noise is a wiener filter, which determines the wiener gain using SNR before and after filtering. The Modified Noise Reduction Method (MNRM) is an innovative approach to reducing feedback noise [15]. On the other hand, research introduced audio classification, robust sound event categorization, and noise using Convolutional Neural Networks (CNN). To account for brief secondary path delays and reduce noise ratio, a fixed-filter noise canceling system employing a convolutional neural network (CNN) classification algorithm was proposed. CNN noise cancellation improved the signal-to-noise ratio (SNR) by 2.3 dB over the adaptive LMS approach. To cancel noise in time-varying systems a frequency-domain noise categorization and coefficient selection approach is proposed [16]. There have been various studies on the classification of noise using CNN [17–19].

However, despite these advancements, there is a notable gap: the limitations of existing Wiener filter and CNN research in evaluating the quality of audio transcriptions to identify noise presence in real-world data. This study aims to fill this gap by focusing on the transcription accuracy of audio quality in the presence of noise using the gathered applicant audio data. We trained our CNN models with clean and noisy audio data to enhance transcription accuracy, thereby addressing the real-world challenges of noise in audio-to-text conversion.

In summary, we observed a research gap in the specific application of enhancing API-based AI models in noisy audio environments. Furthermore, the absence of comprehensive evaluation methodologies, limited exploration of novel noise removal techniques, and a narrow scope of real-world applications further contribute to this gap. To address these shortcomings, this study seeks to establish a strong and specialized technique. that effectively addresses noise removal challenges encountered by API-based AI models in noisy audio environments, thereby contributing to the advancement of noise reduction techniques and the optimization of AI model performance. Thus, the successful implementation of this methodology will lead to significant advancements in noise removal techniques, contributing to more accurate and reliable AI models in noisy audio environments, and benefiting applications such as call centers, voice assistants, and audio analytics.

3. METHODOLOGY

This paper proposes a complete approach for evaluating audio data in noisy contexts to improve API-based AI models. Our method combines signal processing techniques and Convolutional Neural Networks (CNN) to enhance noise removal efficiently. Initially, the audio data is preprocessed, where it is divided into smaller chunks using mel-spectrogram representations from the Librosa library. By applying a Wiener filter to these chunks, researchers successfully obtain cleaner noise versions, effectively mitigating the impact of noise. Audio chunks are classified into 'raw data' (with noise) and 'clean noise' (applied Wiener filter) classes using a CNN model with 1551 mel-spectrogram images on each categorical class. The training and testing data set of each categorical classification split 80% and 20%. As a

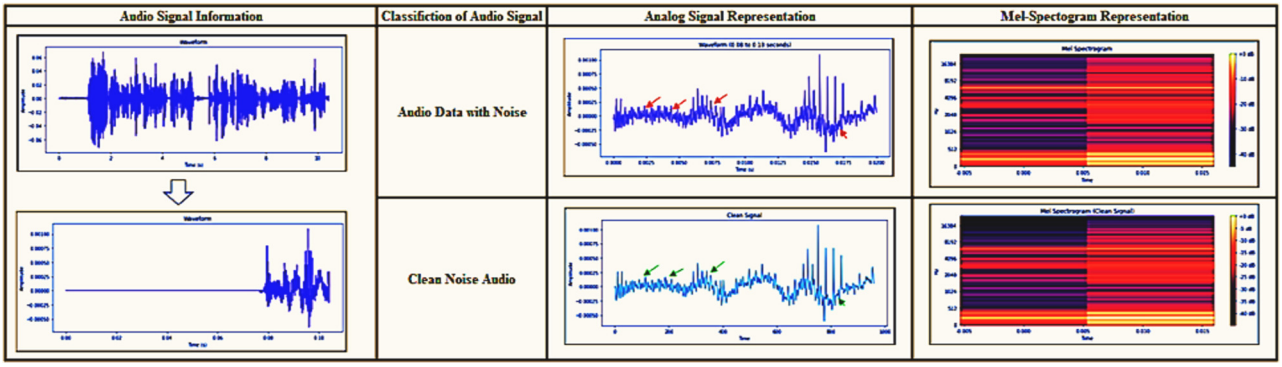


Fig.2: *Audio data cleaning through Wiener Filter.*

result, in the cross-validation set, we utilized 2481 images for training and 621 images for testing. The hold-out validation set used 2232 images for training and 249 images for testing. To enhance the model's generalization capabilities, data augmentation techniques like random rotation and horizontal flip are employed. The researchers adopted a Stratified K-Fold cross-validation strategy, ensuring robust evaluation and validation of the model's performance. Hence, an essential aspect of our research involves the assessment of various optimizers, namely Adam, SGD (Stochastic Gradient Descent), RMSprop (Root Mean Square Propagation), Adagrad, and Adadelta. In addition, it evaluated the learning curve to assess for overfitting on selected optimizers, and optimized epochs, and by conducting a comprehensive comparison where it aims to identify the most suitable optimizer and epoch for our specific audio data analysis task, one that maximizes noise removal efficacy while maintaining model efficiency. To evaluate the effectiveness of our approach, it employs multiple metrics, including accuracy, F1-Score, recall, and ROC AUC. These metrics provide a comprehensive view of the model's performance. In a separate testing set, it was used to validate the trained CNN model through hold-out validation as illustrated in Figure 4. Our research significantly contributes by improving API-based AI models by enhancing noise removal in audio data. The evaluation of different optimizers provides valuable insights, aiding in optimal Hyperparameter selection and model fine-tuning, and the selection of the best optimizers, optimized epoch, optimized batch size, and learning regularization (L2) were implemented. Ultimately, our work promotes the development of more efficient audio analysis systems for API-based AI models and advances noise reduction approaches in real-world applications.

A. Data Collection

The audio data used in this study was obtained from an outsourcing call center provider. The collected audio data consists of various audio calls and data collected in two months (August to September 2023) where collected data was gathered in different calls within the province of Nueva Vizcaya, Philippines. These audio calls serve as the primary dataset for conducting the proposed study and developing the methodology for effective noise removal from audio signals. The use of a two-month timeframe for data collection adds temporal depth to the dataset capturing potential trends in audio patterns throughout that period. This temporal component is important for understanding different noise in audio signals changes from different applicants' background environments, which influences the effectiveness of noise reduction procedures. These audio calls serve as the foundation of the study, providing the basic dataset for the development and testing of the suggested methodology for a successful noise reduction from audio signals.

Hence, using audio data from an outsourcing call center provider serves as a strong foundation for performing a comprehensive study on noise reduction from audio signals, ensuring that the research findings are relevant and applicable to real-world applications.

B. Signal Processing

The audio data is loaded from the specified directory, and each audio file is divided into smaller chunks

Transcription of X applicant.wav: good morning i'm X i want to become a call center agent because i believe that this is what i do best even though not for now but soon i will and i want to work in a field where in i would improve everyday so my expectations to my day to day workload in this type of industry is i'm already i think i know that it will become stressful but no job is easy if you have **ghost** if you have dreams you have to continue with whatever you are doing so and my goals and aspirations in the next five years i want to become more confident with myself and what with i am doing of course as a mother i would also want to provide properly for my children and i think if i got in in this job it will become a big help for me and my family thank you

Fig.3: *Transcribed audio to text.*

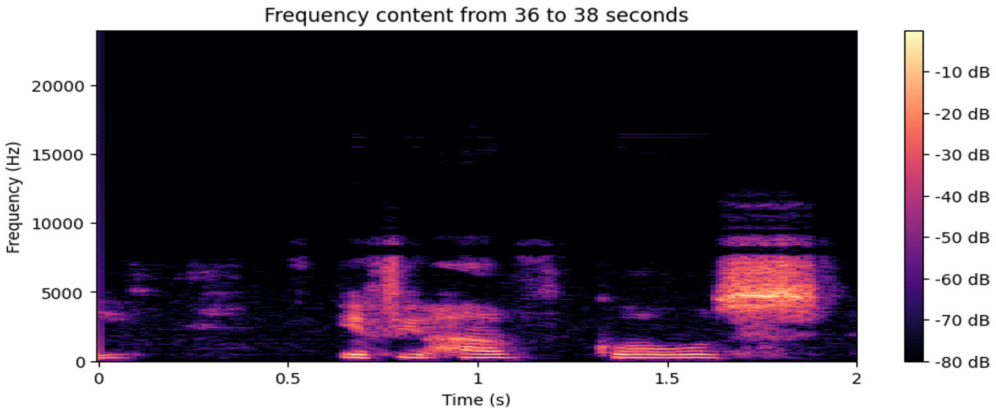


Fig.4: *Noise presence time frame from 36 to 38 seconds – misspelled word.*

with a window size of 2 seconds and an overlapping of 50% (1 second), for each chunk, the mel-spectrogram representation is calculated using the Librosa library, which converts the audio signal into a visual representation. A Wiener filter is applied to obtain a clean noise version of the chunk, which helps to denoise the audio signal where mel-spectrogram representations of the raw data and clean noise chunks are saved as images for further analysis (Figure 2). With this approach, we have collected 3102 Mel-spectogram images.

C. Wiener Filter

One critical aspect of this study is the reliance on automated transcription systems to convert audio interviews into text format and it is crucial for fair and consistent applicant grading. However, noise prevalent in call center environments poses a significant challenge to achieving high-quality audio-to-text transcription. Originating from environmental factors, cellphone microphone interference, and signal transmission distortions, this noise degrades audio signals, leading to transcription inaccuracies and potentially erroneous applicant grading. To evaluate noise's impact, audio samples were transcribed without applying a Wiener filter, revealing significant transcription errors due to noise, as depicted in Figure 3, while Figure 4 illustrates the time frame of noise presence. Researchers audited these samples to identify misspelled words and analyze noise presence, laying the groundwork for developing robust noise mitigation methods to enhance transcription accuracy from applicant audio sets. This involved initializing a speech recognizer and processing each audio file to convert speech to text. Furthermore, a thorough review of these transcriptions revealed misspelled words due to noise interference.

D. CNN Model for Classification Structure

he images of the raw data and clean noise mel-spectrograms are used as inputs with dimensions of 224 x 224 pixels to a Convolutional Neural Network (CNN) model for classification. The CNN model architecture includes Convolutional 2D (Conv2D) layers for feature extraction, MaxPooling2D layers to downsample, and Dense layers for classification, as illustrated in Table 1 and Figure 5.

D.1 Conv2D Layer 1

The Conv2D layer is composed of a kernel size of 3 x 3, padding of 0, strides of 1 x 1, and filters of 32. The calculated output dimension for height and width is 222 x 222.

Formulas:

$$\text{Output}_{height} = \left(\frac{\text{input}_{height} - \text{kernel}_{height} + (2 \times \text{padding})}{\text{stride}}\right)$$

$$\text{Output}_{width} = \left(\frac{\text{input}_{width} - \text{kernel}_{width} + (2 \times \text{padding})}{\text{stride}}\right)$$

Substituting in the values:

$$\text{Output}_{height} = \left(\frac{224 - 3 + 2 \times 0}{1}\right) + 1 = 222$$

$$\text{Output}_{width} = \left(\frac{224 - 3 + 2 \times 0}{1}\right) + 1 = 222$$

$$\text{Param} = \text{kernel}_{height} \times \text{kernel}_{width} \times \text{input}_{channels} \times \text{output}_{channels} + \text{output}_{channels} + 1$$

$$Parameters = 3 \times 3 \times 1 \times 32 + 32 = 320$$

Substituting in the values:

Thus, the Output Shape is (None, 222, 222, 32) with 320 parameters.

D.2 MaxPooling2D, Layer 1, and Dropout Layer 1

The MaxPooling2D Layer 1 and Dropout Layer 1 don't have trainable parameters and the output shape results are the same (None, 111, 111, 32) with a pool size of (2 x 2). The calculated output dimension for

height and width is 111 x 111.
Formulas:

$$\text{Output}_{height} = \left(\frac{\text{input}_{height}}{\text{pool\_size}}\right)$$

$$\text{Output}_{width} = \left(\frac{\text{input}_{width}}{\text{pool\_size}}\right)$$

Substituting in the values:

$$\text{Output}_{height} = \left(\frac{222}{2}\right) = 111$$

$$\text{Output}_{width} = \left(\frac{222}{2}\right) = 111$$

Thus, the Output Shape is (None, 111, 111, 32) with no trainable parameters.

D.3 Conv2D Layer 2

The Conv2D layer is composed of a kernel size of 3 x 3, padding of 0, strides of 1 x 1, and filters of 64. The calculated output dimension for height and width is 222 x 222.

Formulas:

$$\text{Output}_{height} = \left(\frac{\text{input}_{height} - \text{kernel}_{height} + (2 \times \text{padding})}{\text{stride}}\right)$$

$$\text{Output}_{height} = \left(\frac{111 - 3 + 2 \times 0}{1}\right) + 1 = 109$$

$$\text{Output}_{width} = \left(\frac{111 - 3 + 2 \times 0}{1}\right) + 1 = 109$$

Thus, the Output Shape is (None, 109, 109, 32) with 18,496 parameters.

$$\begin{aligned}\text{Param} =& \text{kernel}_{height} \times \text{kernel}_{width} \times \text{input}_{channels} \\ & \times \text{output}_{channels} + \text{output}_{channels} + 1\end{aligned}$$

$$Parameters = 3 \times 3 \times 32 \times 64 + 64 = 18,496$$

D.4 MaxPooling2D Layer 2 and Dropout Layer 2

The MaxPooling2D Layer 1 and Dropout Layer 1 don't have trainable parameters, and the output shape results are the same (None, 54, 54, 64) with a pool size of (2 x 2). The calculated output dimension for height and width is 54 x 54.
Formulas:

$$\text{Output}_{height} = \left(\frac{\text{input}_{height}}{\text{pool\_size}}\right)$$

$$\text{Output}_{width} = \left(\frac{\text{input}_{width}}{\text{pool\_size}}\right)$$

Substituting in the values:

$$\text{Output}_{height} = \left(\frac{109}{2}\right) = 54$$

$$\text{Output}_{width} = \left(\frac{109}{2}\right) = 54$$

Thus, the Output Shape is (None, 54, 54, 64) with no trainable parameters.

D.5 Flatten Layer

The flattened dimension consists of 186 rows and 624 columns. The estimated output flatten layer produced a vector of 186,624, which is represented as (None, 186,624). Where the flattened output size equals the spatial dimensions multiplied by the number of channels.

D.6 Dense Layer 1 and Dense Layer 2

Dense Layer 1 has 128 output units, where the previous layer's input units are 186,624, for a total of 23,880,000. The Dense Layer 2 contains one output unit, while the input layer from Dense Layer 1 is 128, for a total parameter of 129.
Dense Layer 1:

$$\text{Param} = (\text{input}_{Units}) \times (\text{output}_{Units}) + (\text{output}_{units})$$

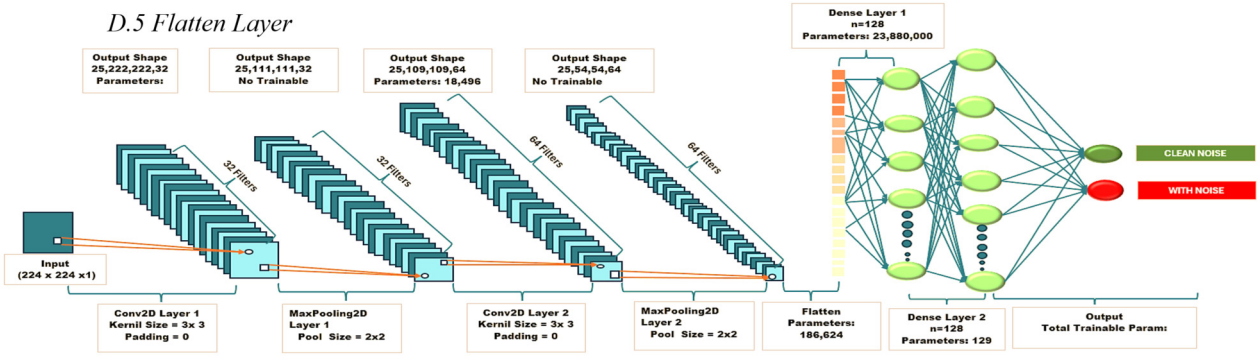


Fig.5: *CNN model Architecture.*

$$\text{Param} = 186,624 \times 128 + 128 = 23,880,000$$

Dense Layer 2:

$$\text{Param} = (\text{input}_{Units}) \times (\text{output}_{Units}) + (\text{output}_{units})$$

$$\text{Param} = 128 \times 1 + 1 = 129$$

In Summary, the CNN model architecture processes clean and noisy images with the shape of 224 x 224 and a grayscale image through CNN and pooling layers, gradually decreasing the spatial dimensions. The Conv2D layers capture features, while Max Pooling layers downsample the data. The dropout layers introduce regularization, and the Dense layers at the end perform the classifications. As a result, the CNN model architecture may capture intricate patterns and characteristics in the image classification class.

Table 1: *CNN Model Summary.*

Model: Sequential		
Layer (type)	**Output Shape**	**Param ♯**
conv2d	(25, 222, 222, 32)	320
max_pooling2d	(25, 111, 111, 32)	0
dropout	(25, 111, 111, 32)	0
conv2d	(25, 109, 109, 64)	18496
max_pooling2d	(25, 54, 54, 64)	0
dropout	(25, 54, 54, 64)	0
flatten	(25, 186624)	0
dense	(25, 128)	23888000
dense	(25, 1)	129
Total Param:		23,906,945
Trainable Param:		23,906,945
Non-trainable Param:		0

E. CNN Model for Classification Structure

The flowchart illustrates a methodology for enhancing API-based AI models in noisy audio environments. It involves signal processing steps, including Mel-spectrogram computation, and Wiener filtering, to obtain clean noise versions of audio chunks. The processed data is then used to train a CNN model for classifying audio chunks in 'raw data (with noise)' and the 'clean noise' categories as shown in Figure 6.

4. RESULTS AND DISCUSSION

In this study, the researcher conducted a statistical evaluation of audio samples to assess the independence of variables representing noise and clean noise. The results revealed a significant difference in the signal-to-noise ratio (SNR) between the two variables, emphasizing the importance of advanced algorithms for effective noise separation [20]. The signal-to-noise ratio is a measure used to quantify the level of desired signal relative to the level of noise. It is defined by the following formula:

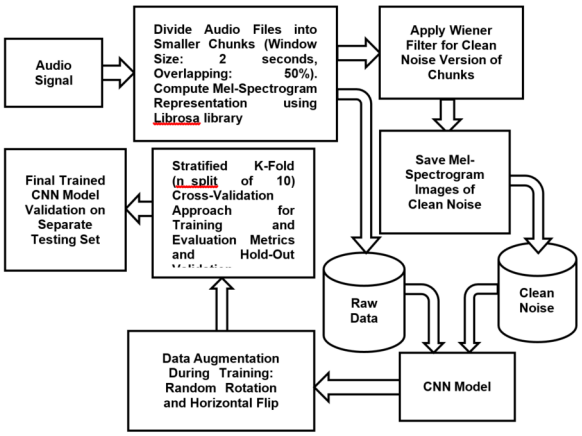


Fig.6: *Methodology Framework.*

Formula:

$$SNR = 10 \log_{10} \left(\frac{\mathrm{P}_{\text{signal}}}{\mathrm{P}_{\text{noise}}} \right)$$

Where $\mathrm{P}_{\text{signal}}$ represents the average power of the signal, and $\mathrm{P}_{\text{noise}}$ represents the average power of the noise.

The box plot clearly illustrates the differences in the SNR values between clean and noise signals, with the clean signal having a wider range and potentially lower SNR values compared to the noise signal as illustrated in Figure 7.

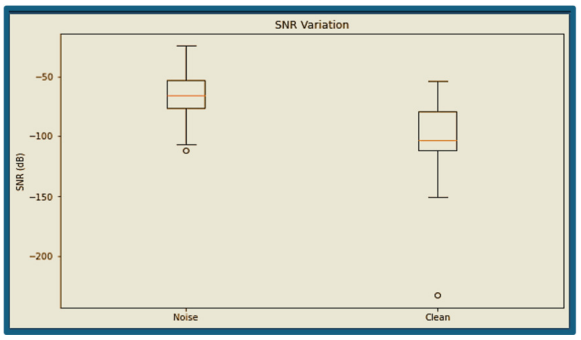


Fig.7: *SNR values between the clean and noise signals.*

Statistical Treatment using Independent T-test:

The statistical treatment involves performing an independent t-test to compare the means of Clean SNR and Noise SNR. The t-test will determine if there is a significant difference between the two groups to be evaluated.

1] Problem Statement:

The problem is to assess whether Clean SNR and Noise SNR are independent and have distinct characteristics. The goal is to determine if the noise signal is stronger compared to the clean signal, and if there is a significant difference in the signal-to-noise ratios between the two groups.

2] Formulate Hypothesis:

2.1] Null hypothesis (H0): there is no significant difference between the means of Clean SNR and Noise SNR

2.2] Alternative hypothesis (H1): there is a significant difference between the means.

3] Confidence Level = 0.95

Significance Level = 0.05

4] Basic Assumptions:

4.1] Independence: Each group's observations are independent of one another.

4.2] Normality: The distribution of data in each group follows a normal distribution.

4.3] Homogeneity of Variance: The variances of Clean SNR and Noise SNR are equal.

5] Statistical Significance:

Higher negative values indicate a poorer signal-to-noise ratio, and a lower quality of the intended signal compared to the noise level. The Clean SNR and Noise SNR have negative values, indicating that the noise signal is stronger than the clean signal. The bigger standard deviation for Clean SNR compared to Noise SNR suggests that the Clean SNR values are more variable, indicating a wider range of signal-to-noise ratios in the clean signal. The calculated means of Clean SNR and Noise SNR are reasonably exact, as indicated by reduced standard errors, which measure the accuracy of the estimated mean values as shown in Table 2.

Table 2: *Summary of Mean, Standard Deviation, and Standard Error.*

	Mean	**Standard Deviation**	**Standard Error**
Clean SNR	-98.829	24.961	1.949
Noise SNR	-66.099	16.945	1.323

The result of the Independent T-test, there is a considerable difference between the means of Clean SNR and Noise SNR, as indicated by the t-statistic of 13.8928014 and the extremely low p-value (7.90E-35). The cleaner signal may have a wider range of signal-to-noise ratios because the Clean SNR Variance (623.054471) is greater than the Noise SNR Variance (287.1440707). These data, combined with 164 degrees of freedom in Table 3, provide strong statistical evidence that Clean SNR and Noise SNR are independent and have distinct properties.

Table 3: *Statistical Summary.*

t-statistic	13.892
p-value	7.90E-35
Clean SNR Variance	623.054
Noise SNR Variance	287.144
DF	164

6] Practical Significance

The significant difference in signal-to-noise ratio and the higher variability observed in the clean signal compared to the noise signal have important practical implications. These findings highlight the need for effective noise reduction techniques and signal processing methods to improve the signal quality and system design in the specific application. Thus, by addressing the significant disparity in SNR, it becomes evident the presence of noise degrades the desired signal, emphasizing the importance of advanced algorithms capable of effectively separating the desired signal from the noise. Additionally, the higher variability in clean signal SNR values underscores the need to account for different signal conditions and develop robust system designs capable of handling varying noise levels.

Throughout the evaluation, the CNN model underwent multiple tests using different optimizers, and epochs to optimize the best accuracy was evaluated in hold-out validation results. The evaluation approach included training and testing the model with an 80%-20% split, performing 10-fold cross-validation, and employing hold-out validation techniques. These rigorous evaluations contributed to the proposed project's success in noise removal in API-Based AI Models.

The CNN model employs a variety of optimizers and evaluates them using cross-validation and hold-out validation. The findings show that Adam outperforms other optimizers, achieving an accuracy of 96.37%, recall of 92.86%, F1-Score of 96.29%, and ROC_AUC of 99.97% for cross-validation, and accuracy of 97.26%, recall of 94.65%, F1-Score of 97.08%, and ROC_AUC of 99.93% for hold-out validation. RMSprop trails closely behind, while Adagrad fares poorly. The evaluation metrics, including accuracy, recall, F1-Score, and ROC_AUC, are presented in Table 4, confirming Adam's superiority as the most effective optimizer for the given batch size and epoch duration.

Moreover, the epoch was optimized, and select different epoch values like 35, 75, 105, and 125 were and epoch 105 was selected with an accuracy of 99.52%, Recall of 100%, F1-Score of 99.50%, and ROC_AUC of 99.99% for cross-validation while Accuracy of 98.79%, Recall of 99.21%, F1-Score of 98.81%, and ROC_AUC of 99.54% for hold-out validation, significantly improving AI model performance as illustrated in Table 5. Also, the confusion matrix resulted in a true positive of 319 as clean and correctly predicted as clean while a false positive of 3 is incorrectly predicted as clean. On the other hand, the true negative of 299 is noise and correctly predicted as noise while the false negative 0 indicates no incorrect predicted as noise (Table 6).

In summary, the classifier describes that the model is performing well with high True positives, and True

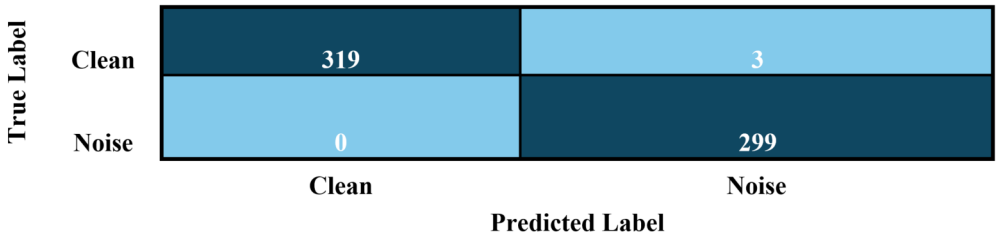
Table 4: *Evaluation of Optimizers Cross Validation and Hold-out Validation.*

Parameters			Cross Validation				Hold-Out Validation			
Optimizer	**Batch Size**	**Epoch**	**Accuracy**	**F1 Score**	**Recall**	**ROC AUC**	**Accuracy**	**F1 Score**	**Recall**	**ROC AUC**
Adam	**25**	**25**	**96.37%**	**96.29%**	**92.86%**	**99.97%**	**97.26%**	**97.08%**	**94.65%**	**99.93%**
Sgd	25	25	50.81%	67.38%	100.00%	52.50%	48.63%	64.91%	98.66%	52.75%
RMSprop	25	25	96.37%	96.29%	92.86%	99.92%	96.94%	96.72%	93.64%	99.99%
Adagrad	25	25	49.19%	14.86%	8.73%	55.35%	50.72%	12.57%	7.36%	50.45%
Adadelta	25	25	52.82%	64.65%	85.00%	52.00%	53.14%	62.26%	80.27%	54.62%

Table 5: *Evaluation of Optimizers Cross Validation and Hold-out Validation.*

	Cross Validation				Hold-Out Validation			
Epoch	**Accuracy**	**F1 Score**	**Recall**	**ROC AUC**	**Accuracy**	**F1 Score**	**Recall**	**ROC AUC**
35	91.13%	91.47%	93.65%	97.22%	90.66%	90.61%	93.65%	97.71%
75	93.95%	94.07%	94.44%	97.82%	92.75%	92.71%	95.65%	98.50%
105	**98.79%**	**98.81%**	**99.21%**	**99.54%**	**99.52%**	**99.50%**	**100.00%**	**99.99%**
125	99.19%	99.21%	99.21%	99.95%	98.87%	98.84%	100.00%	99.99%

Table 6: *Confusion Matrix.*



negatives, and very low False positives, and False negatives, and the selected optimizers and epoch have been optimized to achieve good metrics that can predict the noise and clean noise in API-based as proposed.

To assess the created model, we thoroughly evaluated training accuracy and validation accuracy, learning rate curve, and Receiver Operating Characteristic Area Under the Curve (ROC_AUC) curve to assess if the model is overfitting (Figures 8, 9, 10). On the plotted figures, the training accuracy and validation accuracy are significantly learning without any overfitting as observed, despite the learning curve resulting in the validation loss being larger than the training loss, this will not result in overfitting with a small gap. Additionally, the ROC_AUC curve resulted in 100% hold-out validation indicating a perfect classifier. The model is not overfitting, as evidenced by the training and validation sets of results that indicate a small gap even in the learning curve loss. In addition, to ensure that a batch size of 25 is suitable for our model, we evaluated and optimized different batch

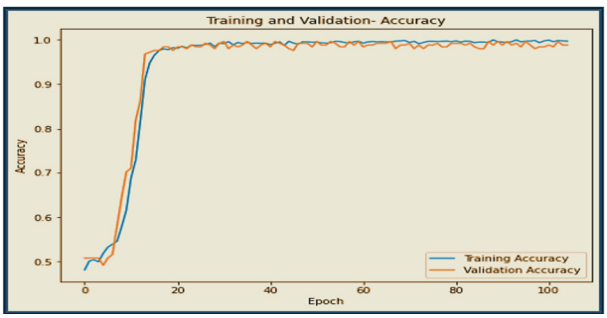


Fig.8: *The training accuracy and validation accuracy.*

sizes (25, 50, 75, and 125) as shown in Figure 11. It was found that a batch size of 25 yields the best accuracy of 99.77% and a validation accuracy of 98.54%. The batch size of 50 converged at epoch 9 with an accuracy of 50.43% and a validation accuracy of 50.81%. A batch size of 75 resulted in an accuracy of 99.64%

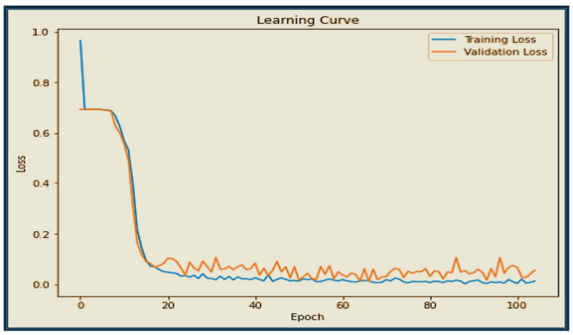


Fig.10: *Learning curve (Training loss and Validation loss).*

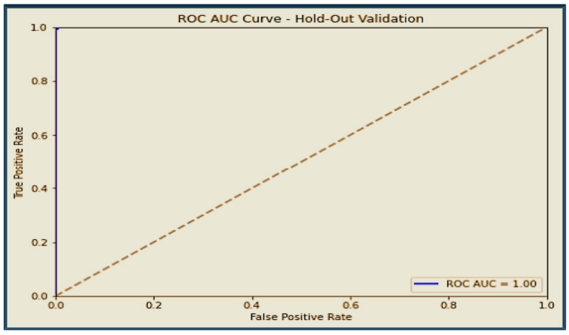


Fig.11: *ROC AUC Curve (Hold-Out Validation).*

and a validation accuracy of 98.39%, while a batch size of 125 achieved an accuracy of 98.97% and a validation accuracy of 98.79%, respectively. This fine-tuning used the optimized optimizer, Adam, and an epoch of 105.

The audio set sampled from the applicant was rigorously analyzed to identify the noise affecting the transcription accuracy, as shown in Figure 12. The researchers identified the background noise impacting the audio quality without using the Wiener filter and corrected the misspelled word using the Wiener filter. The comparison between the original audio (without the Wiener filter) and the enhanced audio (with the Wiener filter) demonstrates a significant improvement in transcription accuracy. In the original audio, noise interference caused the transcription to incorrectly capture the word "ghost" instead of "goals." However, after applying the Wiener filter, the enhanced audio accurately transcribes "goals," showcasing the filter's effectiveness in reducing noise and improving clarity. This enhancement not only results in more coherent and contextually accurate transcriptions but also underscores the importance of noise reduction techniques in ensuring the reliability of AI models that rely on such data. Improved transcription accuracy is crucial for various applications, including automated interview assessments, call centers, and voice-activated systems, where understanding spoken words correctly is essential. Overall, this enhancement highlights the benefits of integrating traditional signal processing methods with modern AI to achieve superior performance in noisy environments.

The implementation of CNN was done using Spyder Python and it was trained and tested on a computer equipped with an 11th Gen. Intel® Core™ i7 processor, and a GeForce RTX™ 3060 Laptop GPU with 6GB GDDR6.

5. CONCLUSION

This research has successfully developed an effective noise removal from audio signals, significantly enhancing the speech quality and intelligibility of API-

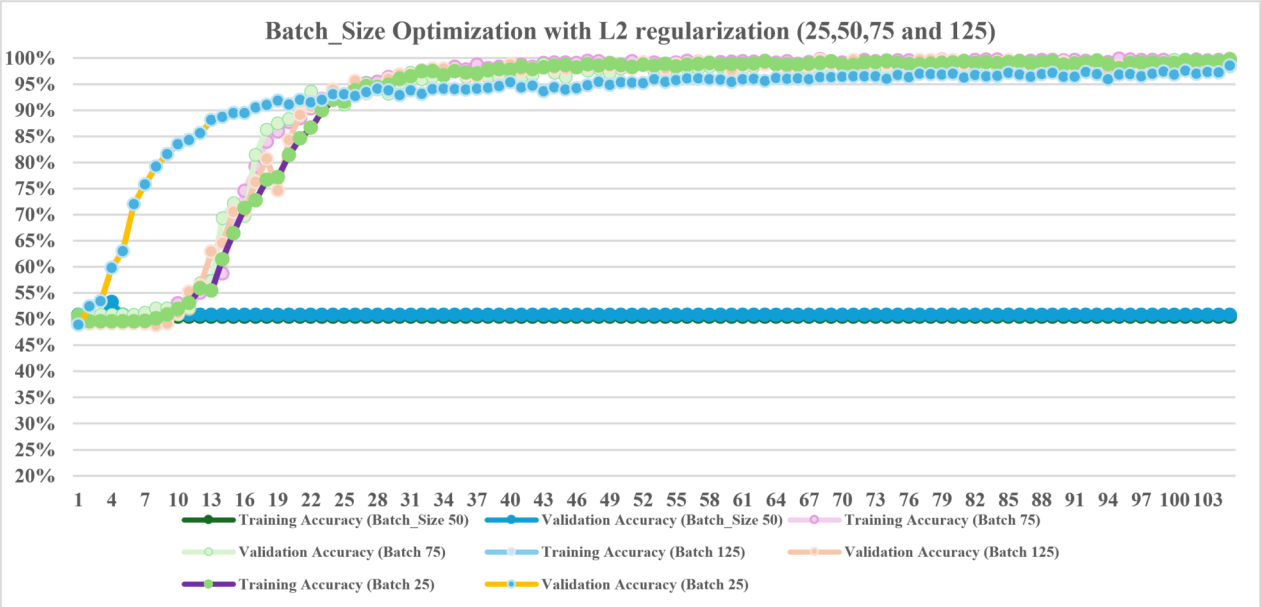


Fig.9: *Batch Size Optimization with L2 regularization (25, 50, 75, and 125).*

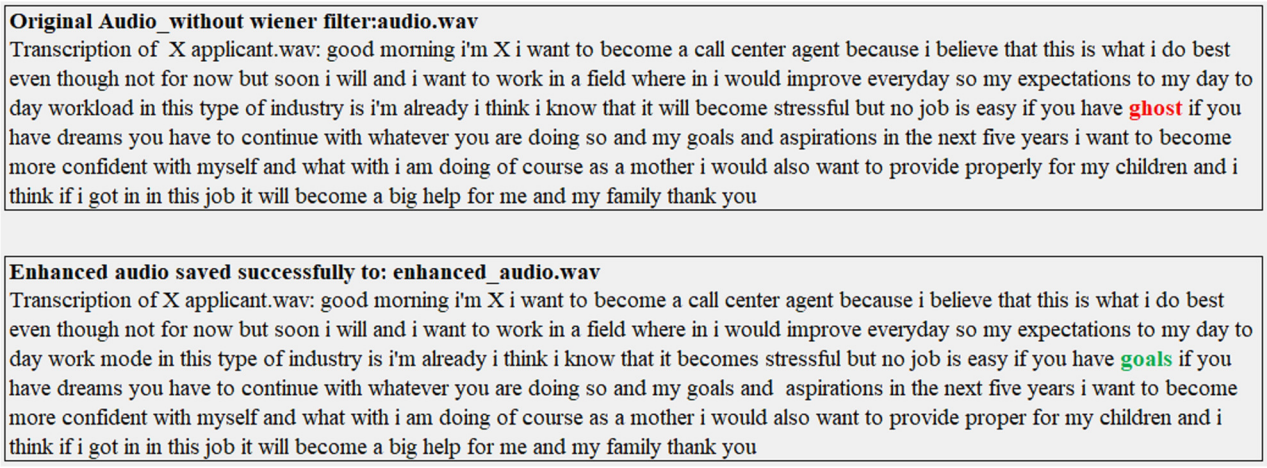


Fig.12: *Applying Wiener Filter.*

based AI models in noisy audio environments and by combining advanced signal processing techniques, including Short-Term-Windowing, SFFT, and Wiener Filter functions with Convolutional Neural Networks (CNNs), the proposed methodology effectively separates clean speech from unwanted noise components. The evaluation of various optimizers, including Adam, SGD, RMSprop, Adagrad, and Adadelta, revealed that the Adam optimizer outperforms others, and optimized the epoch, providing the best results. The utilization of data augmentation techniques further enhanced the model's generalization capabilities. The statistical evaluation of audio samples demonstrated a significant difference in the signal-to-noise ratio (SNR) between clean and noise signals, emphasizing the need for robust noise removal techniques to address varying noise levels effectively. The successful implementation of the methodology holds great promise for real-world applications, particularly in call centers, voice assistants, and audio analytics domains. Overall, this research contributes to the advancement of noise reduction techniques in AI models, fostering optimized performance in noisy audio environments and improving AI model reliability and signal quality, the proposed methodology opens new avenues for enhancing communication, automation, and human-computer interaction fields, unlocking the potential for innovative speech-based applications and services. In conclusion, the innovation of noise removal techniques in real-world, noisy environments significantly enhances the performance and reliability of API-based AI models, paving the way for more efficient and accurate speech recognition and audio data processing.

In future research, we aim to investigate advanced signal processing techniques, explore alternative feature representations, validate robustness with diverse datasets, develop real-time noise removal algorithms, and further enhance the noise removal effectiveness in audio data processing. Techniques such as spectral subtraction, adaptive filtering, noise estimation algorithms, and hybrid Long Short-Term Memory (LSTM) algorithms are integrated to classify a noisy and clean audio signal. Current research acknowledges certain limitations. The effectiveness of noise removal heavily relies on the quality and size of the training dataset, which may impact the model's performance when dealing with rare or specific noise patterns. Despite this limitation, research provides a foundation for effective noise removal in audio data processing, contributing to the innovation of robust solutions for enhancing speech quality and intelligibility in noisy audio environments.

ACKNOWLEDGEMENT

The researchers extend their heartfelt gratitude to all individuals and organizations who contributed ideas and support for this project. Without their generous support and assistance, this endeavor would not have been possible. Throughout this journey, the researchers were fortunate to collaborate with a dedicated team of researchers, mentors, and colleagues. Their invaluable advice and skills significantly improved the quality of their work. Furthermore, the researchers are grateful for the continuous encouragement and understanding from their friends and family, who stood by them during this challenging, yet successful project. The authors also thank the Polytechnic University of the Philippines (PUP) for their assistance and resources. The facilities, financing, and access to essential literature and research resources all contributed significantly to the project's success.

AUTHOR CONTRIBUTIONS

Conceptualization, Joel Ryan De Guzman; Methodology, Joel Ryan De Guzman; Formal analysis, Joel Ryan De Guzman, Robert de Luna, and Marife Rosales; Investigation, Joel Ryan De Guzman; Data

curation, Joel Ryan De Guzman; Writing—original draft preparation, Joel Ryan De Guzman; Review, Robert de Luna and Marife Rosales; Supervision, Robert de Luna and Marife Rosales. All authors have read and agreed to the published version of the manuscript.

References

[1] L. Li, T. Lassiter, J. Oh and M. K. Lee, "Algorithmic Hiring in Practice: Recruiter and HR Professional's Perspectives on AI Use in Hiring," in *AIES 2021 - Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society, Association for Computing Machinery, Inc*, pp. 166–176, Jul. 2021.

[2] L. Armstrong, J. Everson and A. J. Ko, "Navigating a Black Box: Students' Experiences and Perceptions of Automated Hiring," in *Proceedings of the 2023 ACM Conference on International Computing Education Research V.1*, New York, NY, USA: ACM, pp. 148–158, Aug. 2023.

[3] J. S. Black and P. van Esch, "AI-enabled recruiting: What is it and how should a manager use it?," *Business Horizons*, vol. 63, no. 2, pp. 215–226, Mar. 2020.

[4] K. Priya, S. M. Mansoor Roomi, P. Shanmugavadivu, M. G. Sethuraman and P. Kalaivani, "An Automated System for the Assesment of Interview Performance through Audio & Emotion Cues," *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, Coimbatore, India, pp. 1049-1054, 2019.

[5] P. Senarathne, M. Silva, A. Methmini, D. Kavinda and S. Thelijjagoda, "Automate Traditional Interviewing Process Using Natural Language Processing and Machine Learning," *2021 6th International Conference for Convergence in Technology (I2CT)*, Maharashtra, India, pp. 1-6, 2021.

[6] J. S. Queiroz and E. L. Feitosa,"A WebBrowser Fingerprinting Method Based on the Web Audio API," *Gerontologist*, vol. 59, no. 3, pp. 1106–1120, Jun. 2019.

[7] H. Lindetorp and K. Falkenberg, "Putting Web Audio API to the test: Introducing WebAudioXML as a pedagogical platform," 2021. [Online]. Available: `http://nobelprizemuseum.se`

[8] R. B. Ibrahim, "Performance Analysis: AI-based VIST Audio Player by Microsoft Speech API," *Kurdistan Journal of Applied Research*, pp. 21–28, Jul. 2021.

[9] N. Anggraini, A. Kurniawan, L. K. Wardhani and N. Hakiem, "Speech Recognition Application for the Speech Impaired using the Android-based Google Cloud Speech API," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 16, no. 6, p. 2733, Dec. 2018.

[10] R. Cutler *et al.*, "ICASSP 2022 Acoustic Echo Cancellation Challenge," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, Singapore, pp. 9107-9111, 2022.

[11] N. Alamdari, S. Yaraganalu and N. Kehtarnavaz, "A Real-Time Personalized Noise Reduction Smartphone App for Hearing Enhancement," *2018 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, Philadelphia, PA, USA, pp. 1-5, 2018.

[12] S. Liu, G. Keren, E. Parada-Cabaleiro, and B. Schuller, "N-HANS: A neural network-based toolkit for in-the-wild audio enhancement," *Multimedia Tools and Applications*, vol. 80, no. 18, pp. 28365–28389, Jul. 2021.

[13] T. V. and T. R. S. and K. C. and J. P. Senthilkumar Radha and Raghavasimhan, "RealTimeSuppression of Non-stationary Noise for Web-Based Calling Applications," in *ICT Systems and Sustainability, S. and J. A. Tuba Milan and Akashe*, Ed., Singapore: Springer Nature Singapore, pp. 131–140, 2023.

[14] M. A. Kumar and K. M. Chari, "Noise Reduction Using Modified Wiener Filter in Digital Hearing Aid for Speech Signal Enhancement," *Journal of Intelligent Systems*, vol. 29, no. 1, pp. 1360–1378, Jan. 2020.

[15] M. C. R. Kumar and M. P. Chitra, "Implementation of Modified Wiener Filtering in Frequency Domain in Speech Enhancement," 2022. [Online]. Available: `www.ijacsa.thesai.org`

[16] S. Park and D. Park, "Low-Power FPGA Realization of Lightweight Active Noise Cancellation with CNN Noise Classification," *Electronics (Basel)*, vol. 12, no. 11, p. 2511, Jun. 2023.

[17] I. Ozer, Z. Ozer, and O. Findik, "Noise robust sound event classification with convolutional neural network," *Neurocomputing*, vol. 272, pp. 505–512, Jan. 2018.

[18] L. Nanni, G. Maguolo, S. Brahnam, and M.Paci, "An Ensemble of Convolutional Neural Networks for Audio Classification," *Applied Sciences*, vol. 11, no. 13, p. 5796, Jun. 2021.

[19] N. F. Ali, M. Hussein, F. Awwad and M. Atef, "Convolutional Autoencoder for Real-Time PPG Based Blood Pressure Monitoring Using TinyML," *2023 International Conference on Microelectronics (ICM)*, Abu Dhabi, United Arab Emirates, pp. 41-45, 2023.

[20] D. Ribas, A. Miguel, A. Ortega, and E. Lleida, "Wiener Filter and Deep Neural Networks: A Well-Balanced Pair for Speech Enhancement," *Applied Sciences (Switzerland)*, vol. 12, no. 18, Sep. 2022.

Joel Ryan A. de Guzman earned his Bachelor of Science in Electronics and Communications Engineering from the Nueva Vizcaya State University, Nueva Vizcaya in 2009. He is pursuing his Master of Science in Electronics and Communications Engineering (MSECE) at PUP-Manila. His research interests include applications of Artificial Intelligence, Machine Learning, Neural Networks, and Deep Learning. He is currently affiliated in Microchip Technology, Philippines as Sr. II Product Engineer under Wireless Solution Group.

Robert G. de Luna earned his Bachelor of Science in Electronics and Communications Engineering from the PUP, Batangas in 2002. He took his Master of Science in Electronics and Communications Engineering (MSECE) in June 2016 and completed his Doctor of Philosophy in Electronics and Communications Engineering (PhD ECE) at DLSU Manila in 2020. His research interests include applications of Artificial Intelligence, Machine Learning, Neural Networks, and Deep Learning. He teaches BSECE at the College of Engineering, and he serves as the Program Chair of the PUP Graduate School Master of Science in Electronics Engineering Program.

Marife A. Rosales earned her Bachelor of Science in Electronics and Communications Engineering from the University of Batangas, Batangas City in 2006. She took her MS in EE major in Electronics at TUP–Manila in 2018. She is pursuing her Ph.D. in ECE at DLSU–Manila. Her research interests include applications of Artificial Intelligence, Fuzzy Logic, Neural Networks, Computer Vision, and Digital Signal Processing. She is the current Chief of the Center for Futures Training and Advocacy of RISFI at PUP-Manila. Concurrently, she teaches BSECE at the College of Engineering and MSECE at the Graduate School of the same university.