Noise removal from Audio using CNN and Denoiser

Manmohan Dogra, Saumya Borwankar and Jayashree Domala

Abstract As there is aggrandizement in the sector of artificial intelligence relating to the speech domain, it becomes a necessity to have efficient noise removal models with greater efficiency and less complexity. The presence of noise in audio signals poses a great complication when working on speech recognition, enhancement, improvement, and transmission. Hence there is a necessity to develop the most efficient algorithm for noise reduction which works in real-time and is successful in removing maximum noise. To be above this difficulty, this paper presents an efficient algorithm for noise detection which works on the principles of deep learning specifically CNN (convolutional neural networks) and the removal of similar noise from the audio using the python module 'noise reducer'.

1 Introduction

Noise removal is a process that deals with removing noise from a signal [9]. This process can be used for both images and audios but here the focus is on noise removal for audios. All the signals have characteristics that make them vulnerable to noise. This noise could be white noise [11] or random noise with an even frequency distribution or frequency- dependent noise. The noise could be in the form of but not limited to hum and hiss, drillings, air conditioner, or car horn. If the audio has an unwanted and unpleasant sound then it becomes difficult to use it further for

Manmohan Dogra
St. Francis Institute of Technology, Mumbai - 400103 Maharashtra, e-mail: mohanqwerty5@gmail.com

Saumya Borwankar
Nirma University, Ahmedabad - 382481 Gujarat, e-mail: 17bec095@nirmauni.ac.in

Jayashree Domala
St. Francis Institute of Technology, Mumbai - 400103 Maharashtra, e-mail: domalajayashree@gmail.com

transmission, recognition, or improvement. Considering the various edge devices like phones, laptops, conferencing systems, etc, traditional noise suppression methods have been implemented. Since it is an edge device such an approach seems intuitive. It captures the user's voice in the first place and then device filters are used to noise out the captured voice. Later the output is saved [6]. Another common and consistent solution for controlling or suppressing background noise is sound masking. It adds the white noise commonly referred to as the pink noise into an environment to mask the unwanted sound. But this method is not an active noise control, that is it only reduces or eliminates the perception of noise [10].

As a deduction to this issue at hand, "Noise removal from audio using CNN and denoiser" is introduced as an affirmative solution to solving the problems that arise due to the presence of noises in the audio. A broad goal that we wish to accomplish is noise removal to the bare minimum with the accessibility of doing it in real-time. The core of the project is based on the deep learning principles of CNN (convolutional neural networks). CNN is a special type of artificial neural network (ANN) which has the ability to pick the patterns and make sense out of it. These networks take advantage of local spatial coherence which helps them have shared parameters leading to fewer weights thereby forming convolutions [8]. This aids in extracting relevant information and reduces the computational cost. The system will apply the deep learning algorithm to detect the type of noise in the audio and eventually a 'noise reducer' will be used to remove that particular noise to finally produce a noise-free audio output.

2 Relevant Work

Various researchers have worked on noise reduction techniques utilizing different methodologies. The authors Chavan O., et al., have worked on noise reduction with the help of the audacity noise removal. This is an open- source audio program. The decrease of clamour was achieved by using the Fourier analysis to improve the audio frequency spectrum. The spectrum of pristine tones that make up the background noise are found in the susurration segment [7]. In another paper by Al-Allaf, et al., the authors have made four ANN models. These models are namely Function Fitting (FitNet), Recurrent (RNNs), Cascaded-ForwardNet and Nonlinear AutoRegressive (NARX). They have prepared them independently to turn into a channel to dispose of the noise from any speech signal. From tests led, the best outcomes were gotten from NARAX and FitNet models. But the best algorithm in terms of training, in this case, is TrainLM [4]. While in the paper written by Zoican S., the author presents a framework dependent on the wavelet approach with a level-dependent threshold that gives a valuable technique to eliminate noise in the signals. The test results show that this technique eliminates noise altogether and the framework works progressively [19]. The authors Prasadh S.K., et al., have utilized the strategy where the noise signal is passed through a channel or transformation. Signal Dependent Rank Order Mean algorithm (SD-ROM), which eliminates noise

from audio signals is used. It also retains the characteristics of the signal. To wipe out white Gaussian noise, a discrete wavelet transform method is used. After every one of the methods are applied to the examples, SNR and elapsed time are calculated. The entirety of the above methods show an increased Signal to Noise Ratio (SNR) after processing [17]. A paper by Godsill S., et al., the authors have introduced a signal separation- based way to deal with this issue. Noise transients and audio signals are displayed as autoregressive (AR) measures which are superimposed additively to give the observed waveform. A more practical scheme is then evolved which uses a Kalman filter for the implementation of the separation. The excitation variance of the noise transient model is tightened exponentially to zero away from the irregularity, to evade low-frequency distortions to the sound signal. The strategy is totally mechanized and moreover pragmatic to actualize when contrasted with the current plans for the removal of such deformities. Results indicate a high level of performance [14]. Authors Germain F.G., et al., present a deep learning approach to deal with the process of diminishing the noise in the speech signals by processing the crude waveform directly. The framework plans to deliver a processed signal that contains only the speech content for a given input audio. This input audio would contain speech tainted by an additive noisy background signal. A fully-convolutional context aggregation network is trained using a deep feature loss. This deep loss feature loss is based on comparison. The juxtaposition is done between the internal feature activation in a different network, trained for acoustic condition identification and domestic audio tagging [13]. The writer's Murphy J., et al., gives a technique for the disturbance of commotion which includes nonGaussian impulse from a signal. Gabor relapse version helps in the removal of impulse noise. The parameters are calculated by Gibbs MCC sampler of problem formulated by the Joint Bayesian Framework. Signal to noise ratios improved and the audio quality of the result improved by modeling impulses with discrete switching process.[16][18]. While in another paper the author Ali M., et al., used a hierarchical threshold algorithm which presented noise removal based on double-intensity dual-tree wavelet transform(DDDTWT). Additive White Gaussian Noise mixed with the audio signal is selected for the implementation. The results produced by global threshold method signal to noise ratio, DDDTWT, root mean square error (RMSE) are observed and juxtaposed[5]. Considering the previous research already done in this area, the proposed system focuses more on deep learning based technique specifically using CNN. The main goal is to implement a more convenient and sophisticated system which is reliable.

3 Implementation

The implementation process starts by taking in the input noisy audio which goes through the deep learning model to detect the type of noise present in the input audio. Then the input noisy audio and the detected noise type are passed through a "noise reducer" module of python which damps the noise from the original noisy audio to give a noise suppressed audio with less audio turbulence. The noisy audio dataset

used for training the Neural Network is by 'UrbanSound8k' 6GB in size containing 8732 sound excerpts which are labelled of from 10 most common noise classes: children, air conditioner, dog bark, drilling, car horn, engine running, jackhammer, gunshot, street music and continuous siren. The pseudo-code for implementation is as follows:

Algorithm 1 Model creation

Require: N is the total number of noisy training audios
procedure MATRIX(noisy audio) ▷
    $A1 \leftarrow MFCC(audiofile)$ ▷ Generate MFCC array A1
    $A2 \leftarrow Melspectrogram(audiofile)$ ▷ Generate mel spectrogram array A2
    $A3 \leftarrow chromaSTFT(audiofile)$ ▷ Generate chroma STFT A3
    $A4 \leftarrow chromaCQT(audiofile)$ ▷ Generate chroma CQT array A4
    $A5 \leftarrow chromaCENS(audiofile)$ ▷ Generate chroma CENS array A5
    $S1 \leftarrow A1, A2, A3, A4, A5$ ▷ Stacked to form array S1
end procedure
for `i in range N` **do**
    `append Si to Sn`
    `Repeat the procedure from step 1`
end for
for `i in range N` **do**
    `Si converted to one-hot encoded vector Pi`
    `Vi padded to form Pi`
    `Pi reshaped to form Ri`
end for
procedure MAKE MODEL ▷ Creation of model
    while layers **do**
        $model \leftarrow model.layers$ ▷ Defining model architecture
    end while
    $model \leftarrow model.compile$ ▷ Compiling the model
    return model
end procedure
procedure TRAINING ▷ Model is trained on the vectors Ri and clustered in classes Ci
end procedure
save $model.h5$ ▷ This model file can be used to make predictions
procedure LOADING MODEL ▷ Model M is loaded with noisy input audio X1
    `X1 undergoes from step 1 to step 17 giving finally padded P1`
    `P1 is passed through Model M, which predicts the noise class C1 and C2`
    `Noise N1 of type C1 and N2 of type C2 is fetched from noise samples.`
end procedure
procedure NOISE REDUCER ▷ Using the library noise reducer to remove noise
    `noise reducer(X1,N1)`
    `noise reducer(X1,N2)`
end procedure

This implementation is broken down into 3 modules. These modules are mainly pre-processing, training, and noise removal.

3.1 Module 1. Pre-processing

This module deals with extracting the relevant information from the noisy audio which is later used for training purposes. The audio has five different features on which we mainly focus. These features are:

1. MFCC: Mel Frequency Cepstral Coefficients - It has 39 capabilities, the function be counted is small enough to pressure us to examine the records of the audio. 12 parameters are associated with the amplitude of frequencies. It affords us with sufficient frequency channels to research the audio[15].
2. Mel spectrogram: Compute a mel-scaled spectrogram[3] - It samples the input with window size, computes the Fast Fourier Transform for each window from time domain to window domain. Takes the entire spectrum and evenly separates it into spaced frequencies, and generates spectrograms [12].
3. Chroma STFT: Chroma short term fourier transfer - Compute a chromagram from a waveform or power spectrogram using short term fourier transforms [2].
4. Chroma CQT: Constant-Q Transform - Like mel-scale, the regular-Q transform uses a logarithmically spaced frequency axis, where the window period is distinctive for every frequency. i.e low frequency for long windows and high frequency for short window [1].
5. Chroma CENS: Chroma energy normalized statistics: The CENS features takes statistics facts over huge windows smooths local deviations in tempo, articulation, and musical embellishes inclusive of trills and arpeggiated chords. CENS are best used for tasks such as audio matching and similarity [1].

The more number of features, the better is the collection of data for training the model. Therefore, these five features are taken into account.

The below steps go around in a loop as long as the audio is inputted.

1. Step 1: Input audio. We get the noisy-audio input which is in the “wav” format.
2. Step 2: Generate feature array The 5 features mentioned above are extracted from the noisy audio and are stored in an array. These arrays generated for each noisy audio is further saved in a csv file.

3.2 Module 2. Training

This module deals with the core deep learning algorithm which detects the type of noise present in the input audio.

1. Step 1: Retrieve the amalgamated array from the csv file - The csv file which has each input audio’s extracted features information is used. The csv file is retrieved in the form of a dataframe.
2. Step 2: Convert to categorical features - The algorithm works efficiently if the input values are converted to categorical values. This is done by using one hot encoding. Each array is converted into a one hot encoded vector.

3. Step 3: Padding - The next step is to apply padding to these one hot encoded vectors. This is done to make the length of the vectors equal.
4. Step 4: Reshaping - Since the deep learning algorithm used is convolutional neural networks (CNN), it needs to be reshaped into the 2D form. This final matrix is then stored in the data frame.The matrix in 2D form is retrieved and then converted in 3D form before training.
5. Step 5: Building the model - The model used has 200 input variables. Moreover, it is formed by two 'Convolutional' layers: first with 128 neurons and the second layer with 256 neurons, both with a kernel size of 5, the number of strides as 1 with 'relu' as the activation function. Then following are the two 'MaxPooling' layers with the same padding. Then a dropout layer of 0.3 removing 30% of neurons, followed by a flatten layer. The flatten layer has 5120 variables. The flatten layer is followed by a hidden(dense) layer with 512 neurons and 0.3 dropout layer, then a subsequent dense layer with 1024 neurons and 0.3 dropout layer. In the end, an output layer of 10 categorical output is set. A total set of trainable parameters (3,980,298) was compiled using 'adam' optimizer and 'categorical_crossentropy' as the loss function. The training took place with a batch size of 50 and 40 epochs. The layers of the CNN architecture can be seen in Figure 2.
6. Step 6: Training the model - The last step is the training of the model. The model is trained on the 'train' and 'validation' sets of the data frame and passed through the convoluted neural network. The trained model is saved for future purposes of testing. The flowchart representing pre-processing and training steps is shown in Figure 1.

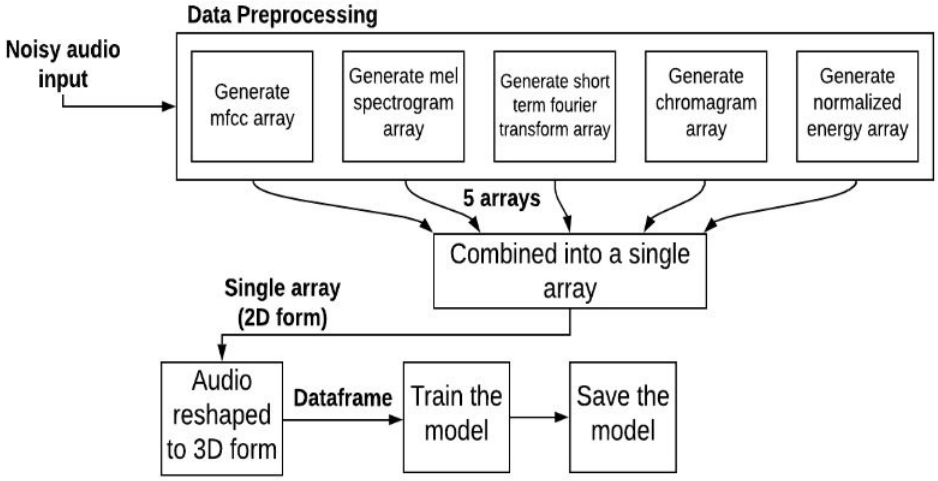


Fig. 1 Flow diagram of the pre-processing and training module

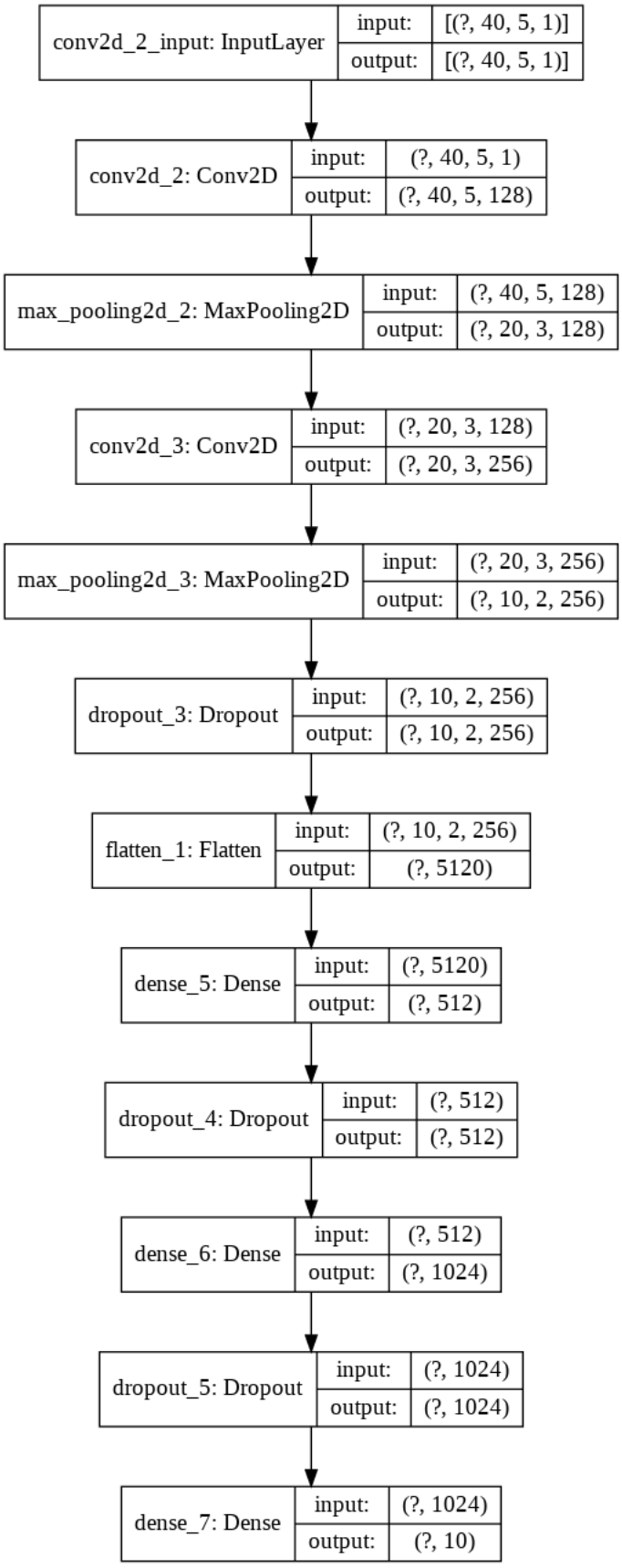


Fig. 2 Representation of the layers of the CNN architecture

The problem with reduction of noise is the recognition of noise, if the noise signal is directly available to us it becomes very easy to remove the noise, but in real case scenarios the noisy audio is not present separately instead it is mixed with the target input which creates a problem to localize the noise. Our method helps tackle this problem as it provides us a way to recognize the noise clip before hand and later remove the noise with the help of two separate clips first being the noise + original audio clip and the second being the noisy audio clip. So this helps us convert noisy input to a denoised signal with the help of steps described in Module 3 Step 4, noise reduce library of python helps us achieve the last task.

3.3 Module 3. Noise removal

This module deals with the final de-noising part.

1. Step 1: Input audio and load model - The previously trained model and the audio input is loaded.
2. Step 2: Pre-processing - The input audio goes through the pre-processing steps as mentioned in module 1. This preprocessing step extracts the features of audio and presents in the 3D vector form.
3. Step 3: Noise type detection This 3D vector is then sent through the trained model. This model outputs the type of noise detected. The detection takes place by comparing a predefined noise dataset with the input noisy audio. This comparison helps to retrieve the types of noise present in the audio.
4. Step 4: Denoiser After the successful detection of the noise, the detected type-of-noise and original input audio is sent through the "noise-reducer" python module. This module removes the noise from the audio and generates a less turbulent audio having reduced noise. The python module is able to do certain transformations, firstly FFT is calculated on the noise clip after which the statistics are computed over the FFT of the noise. Next a threshold is set according to the statistics of the noise. Now the FFT of the signal is calculated, after which a mask is calculated by the comparison of the threshold to the signal FFT. Next the mask is smoothed with a filter over time and frequency. Lastly the mask is put on the FFT of the signal and is inverted.

4 Results

Evaluation metrics play an important role to ascertain the reliability of the model. For the CNN model, the accuracy and loss can be determined which reflects the efficiency of the model. The CNN model gives the training accuracy as **95.82%**, training loss as **0.13**, validation accuracy as **73.59%** and validation loss as **1.02** as shown by the graphs in Figure 4.

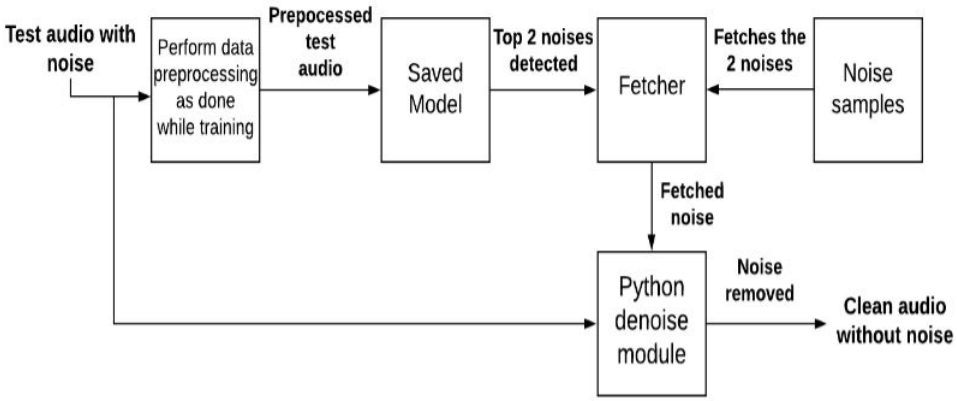
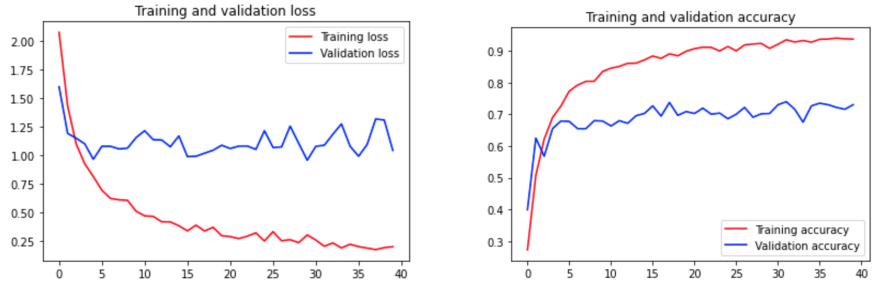


Fig. 3 Flow diagram of the noise removal module

To check the model output 2 samples of noisy data are taken and passed through the CNN model after which the noise reduction takes place in the noise reduction package. Figure 5 represents a noisy input with a street background noise. The desired output along with the output generated by the system are also shown. Similarly, Figure 6 represents a noisy input with children's background noise along with the desired and generated output.



[a] [b]

Fig. 4 Training and Validation (a) Loss and (b) Accuracy on dataset

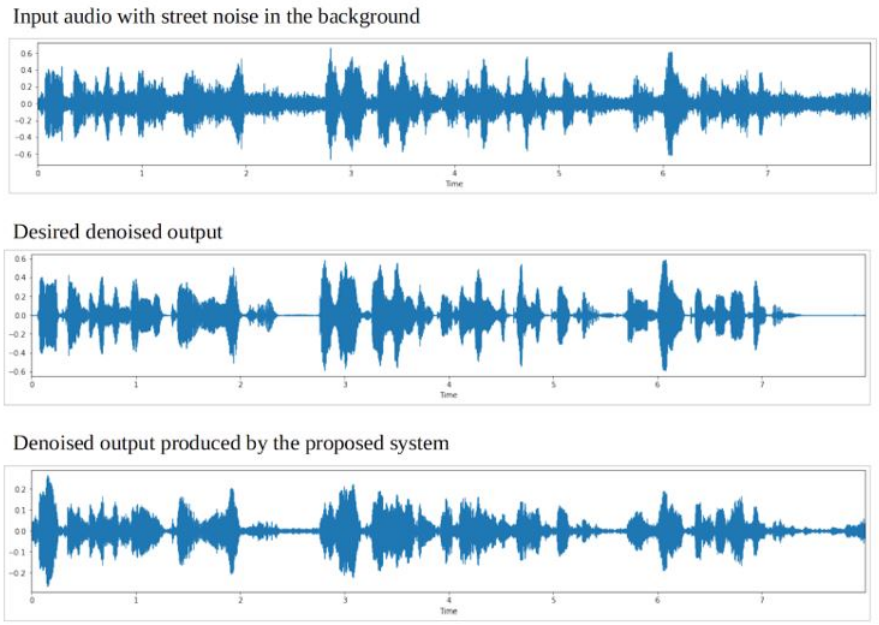


Fig. 5 Noise reduction output 1

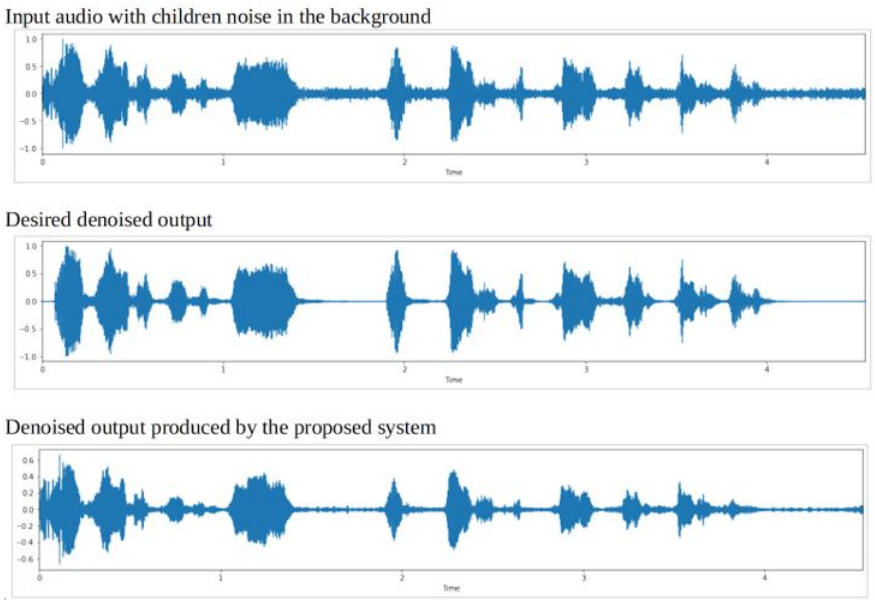


Fig. 6 Noise reduction output 2

5 Conclusion

Through this paper, a system for noise removal from the audio was presented. In particular, a deep learning-based model of CNN was used to detect the noises and then perform the noise removal from audio. The primary research aim was to build an extrinsic and efficient model that can be further incorporated for any audio-related use case. The experimental results shows excellent performance of the model which achieved an accuracy of 97.1%.

For the future scope of this project, an increase in the number of features to get more accurate results and applying HQ filters to mask the noise in audio and prevent audio-loss can be done.

References

1. Constant-q transform and chroma. URL https://musicinformationretrieval.com/chroma.html
2. librosa.feature.chroma_stft. URL http://man.hubwiz.com/docset/LibROSA.docset/Contents/Resources/Documents/generated/librosa.feature.chroma_stft.html
3. librosa.feature.melspectrogram. URL http://man.hubwiz.com/docset/LibROSA.docset/Contents/Resources/Documents/generated/librosa.feature.melspectrogram.html
4. Al-Allaf, O.: Removing noise from speech signals using different approaches of artificial neural networks. International Journal of Information Technology and Computer Science **7**, 8–18 (2015). DOI 10.5815/ijitcs.2015.07.02
5. Ali, M.A., Shemi, P.M.: An improved method of audio denoising based on wavelet transform. In: 2015 International Conference on Power, Instrumentation, Control and Computing (PICC), pp. 1–6 (2015)
6. Baghdasaryan, D.: Real-time noise suppression using deep learning (2020). URL https://developer.nvidia.com/blog/nvidia-real-time-noise- suppression-deep-learning/
7. Chavan, O., Kharade, N., Chaudhari, A., Bhalke, N., Nimbalkar, P.: Machine learning and noise reduction techniques for music genre classification. Machine Learning **6**(12) (2019)
8. contributors, W.: Convolution; wikipedia, the free encyclopedia (2020). URL https://en.wikipedia.org/w/index.php?title=Convolutionamp;oldid=97515 1751. Online; accessed 6-September-2020
9. contributors, W.: Noise reduction, wikipedia the free encyclopedia (2020). URL "https://en.wikipedia.org/w/index.php?title=Noise_reductionamp;oldid=967676491". Online; accessed 6-September-2020
10. contributors, W.: Sound masking; wikipedia, the free encyclopedia (2020). URL https://en.wikipedia.org/w/index.php?title=Sound_maskingamp;oldid=9 65358994. Online; accessed 6-September-2020
11. contributors, W.: White noise;wikipedia, the free encyclopedia (2020). URL https://en.wikipedia.org/w/index.php?title=White_noiseamp;oldid=9755 42178. Online; accessed 6-September-2020
12. Gartzman, D.: Getting to know the mel spectrogram (2020). URL https://towardsdatascience.com/getting-to-know-the-mel- spectrogram-31bca3e2d9d0
13. Germain, F.G., Chen, Q., Koltun, V.: Speech denoising with deep feature losses (2018)
14. Godsill, S.J., Tan, C.H.: Removal of low frequency transient noise from old recordings using model-based signal separation techniques. In: Proceedings of 1997 Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 4 pp.– (1997)
15. Hui, J.: Speech recognition-feature extraction mfcc amp; plp (2019). URL https://medium.com/@jonathan_hui/speech-recognition-feature- extraction-mfcc-plp-5455f5a69dd9

16. Murphy, J., Godsill, S.: Joint bayesian removal of impulse and background noise. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 261–264 (2011)
17. Prasadh, S.K., Natrajan, S.S., Kalaivani, S.: Efficiency analysis of noise reduction algorithms: Analysis of the best algorithm of noise reduction from a set of algorithms. In: 2017 International Conference on Inventive Computing and Informatics (ICICI), pp. 1137–1140 (2017)
18. Wolfe, P., Godsill, S., Ng, W.J.: Bayesian variable selection and regularization for time-frequency surface estimation. Journal of the Royal Statistical Society Series B **66**, 575–589 (2004). DOI 10.1111/j.1467-9868.2004.02052.x
19. Zoican, S.: Audio signals noise removal real time system. pp. 25 – 28 (2010). DOI 10.1109/ICCOMM.2010.5509098