Denoising the Noisy Signal in Speech

Hemant Vardani, Yash Agrawal, Kshitij Garg
Indian Institute of Information Technology Kota

Dr. Isha Pathak Tripathi
Indian Institute of Information Technology Kota

***Abstract*—In the domain of signal processing, speech enhancement is a field that aims to improve the quality of speech by removing the noises present in it. These noises are undesired and might have originated from a surrounding event occurring at the same time, which got superimposed with the speaker's sound. This paper revolves around enhancing speech by reducing the noises present in it. Speech audio can have noises, for instance, horn noise, dog barking, or the noise of a siren. Hence, the paper is presenting an idea to remove them and output the clean audio which no longer has that noise. The Paper talks about various methods which are based on different techniques and methodologies. Those involve machine learning concepts of supervised and unsupervised learning to denoise the audio. These methods were tried on various input audios which had different noises and the results from those were observed (which is shown in the result section of this paper).**

***Index Terms*—Denoising, Speech Enhancement, LSTM, CNN**

I. Introduction

While dealing with the processing of analog signals, there is an overhead of removing undesired signals which got superimposed over the actual signal. For instance, when the sound of the speaker is being recorded, due to the presence of surrounding noises the original signal gets distorted and is unable to convey the complete information to the other side. Therefore, in the contemporary world, the need for audio denoising is huge, be it speech or music restoration. It helps to hear the voice of the desired speaker from the mixed audio instead of so many unwanted noises that get added to the audio. Usually, when audio signals are being transmitted from our device to another device over a distance by any means, the addition of noises to the original signal is observed. This affects the information that is stored in the original signal. So to reduce the noise, an efficient way is needed. [1]

There are many important applications for denoising audio. One such is while online meeting through any meeting software, where the background noise of the person also gets transmitted to another side. This makes it difficult for the receiver to comprehend the actual message. Other can be automatic machines that take input from the person through what he says. If due to noise present in surrounding noise, the information perceived by the machine gets wrong, the result can be devastating. This can be prevented by denoising the input sound.

Reducing noise in a noisy speech recording seems like a difficult task because it is under research for a long time. Researchers have made many improvements in techniques that were used in the 20th century to enhance speech.

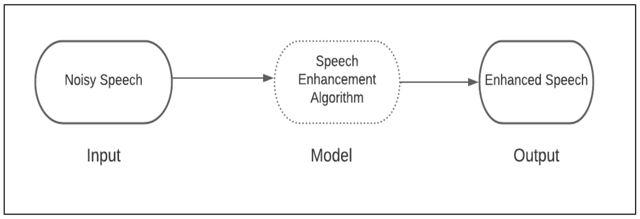

Fig. 1. Basic Block Diagram

To enhance the speech, the fundamental architecture (Fig 1.) usually remains the same. More complex algorithms are introduced to this architecture to improve on. As can be seen, we pass noisy speech as our input then apply our speech enhancement algorithm to get the desired enhanced speech as output as shown in the above block diagram.

Since the inception of this problem statement, the fundamental way is being used is by using threshold frequency which denies all frequencies above it[2]. Surprisingly this way which seems very trivial does perform well in some cases. Still, for other cases, more techniques are required to be discovered. Many better de-noising techniques have been proposed in past for the removal of these noises from a signal.

II. LITERATURE SURVEY

Speech enhancement via removing the noise present in audio remained part of the interest of many scholars. With time, it is seen that these scholars derived many different methods to solve this problem. Sometimes, many methods are there that perfectly remove a particular noise, but fail to give satisfactory results when a sample audio sample is mixed with a different kind of noise.

Researchers have identified noises and classified them to make further denoising of those noises easier. For some noises like gaussian noise, white noise, colored noises, etc the methods to remove them are well known. The speech enhancement (denoising the audio sample) is so deep problem because it is not possible every time to differentiate the present noise into one category, hence those methods have limited application.

Nevertheless, some of the ways which have shown great outcomes are discussed below:-

A. WPT Algorithm

WPT(Wavelet Packet Transmission) is one of the algorithms which denoise the audio signal and output the clean audio as result. This algorithm operates on wavelet analysis. It demonstrates the signal as the sum of multiple wavelet functions. It's been observed that wavelet packet transform outperforms many other algorithms when the application is related to ultrasonic signals and vibration signals.

Using WPT (Wavelet Packet Transmission) audio is decomposed into various coefficients. These coefficients are thresholded with some rules. All these coefficients are collected with inverse packet transform which helps to obtain the denoised signal. [3][4] The simplest model for additive noise has a form f'(k)= f(k) + n(k) , where f'(k)=contaminated signal, f(k)=original signal, n(k)=Noise signal, noise n(k) is considered as white gaussian noise with a 'Zero' mean.

B. CLASSIFICATION AND DENOISING

This way states that the solution to the above problem is firstly, to classify the noises present in audio. This would tell which noises have to be removed from the input audio. Hence here denoising occurs by removing that identified noise from a sample. Using a Convolution Neural Network actually reduces the dimension and gives a label of unwanted noises. [5]

1) Feature Extraction: It extracts some important features of noises named as:
1] MFCC [6]
2] MelSpectrogram [7][8]
3] Croma and STFT [9]
4] Croma and CQT [10]
5] Croma and CENS [11]
By this five features we make a matrix of 1*5 and apply CNN for all audio signal.

2) CNN: CNN applied in the way that supposed to provide the above matrix to all the layers of our architecture. [12]

Conv2D : Conv2D is a layer that convolves over the input and presents the 2D output. It uses a kernel to do this, which can be of any dimension. The value of each cell is a sum of (inputs * weights) which is shifted each time by 1 step horizontally and vertically. [13]

Max Pooling layer : This layer carries on only that information, which is maximum in its local area. Kernel size can be variable, hence the local area can be decided. Popular kernel sizes are 3*3, 5*5, and 7*7. This happens when kernel convolves over the given input data matrix.

Dense Layer : A dense Layer is a layer of neurons in which the inputs of each neuron are all the neurons of the preceding layer. That's why it is called a dense layer. A dense layer is used for changing the dimensions of the vector using matrix chain multiplication. All the extracted features from previous layers are combined to give the final output for classification.

This architecture[14] (Fig. 2) shows initially has an input of (None, 36, 5, 64) like having a matrix

batch-size	epochs			loss	accuracy	items per epochs
50	10	Training	247	0.2943	0.9045	158
		Validation	27	0.9227	0.6846	
50	20	Training	247	0.1775	0.9431	158
		Validation	27	1.1245	0.6918	
50	30	Training	247	0.0909	0.9709	158
		Validation	27	1.5009	0.6583	
50	40	Training	247	0.066	0.9804	158
			27	1.1597	0.7192	

TABLE I
TREND OF CHANGE WITH EPOCH SIZE

batchSize	epochs			loss	accuracy	itemsper epochs
10	40	Training	247	0.1478	0.9597	790
		Validation	27	1.2479	0.6559	790
20	40	Training	247	0.0935	0.9749	395
		Validation	27	1.3645	0.6452	395
40	40	Training	247	0.0728	0.9771	198
		Validation	27	1.2632	0.7133	198
60	60	Training	247	0.0402	0.9885	132
		validation	27	1.2409	0.7049	132
100	100	Training	247	0.0227	0.9927	79
		Validation	27	1.4213	0.7168	79

TABLE II
TREND OF CHANGE WITH BATCH SIZE

of 36*64 for each feature. So a total of five features : (36, 5, 64) for all audio signals so (None, 36, 5, 64) passes into our CNN architecture and gives output as (None, 10), here 10 represents the total of 10 noise labels.

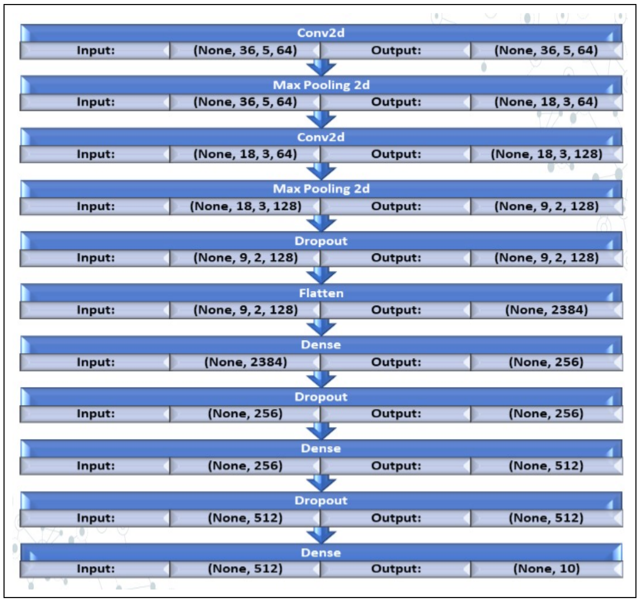
Conv2d
Input: (None, 36, 5, 64) Output: (None, 36, 5, 64)
Max Pooling 2d
Input: (None, 36, 5, 64) Output: (None, 18, 3, 64)
Conv2d
Input: (None, 18, 3, 64) Output: (None, 18, 3, 128)
Max Pooling 2d
Input: (None, 18, 3, 128) Output: (None, 9, 2, 128)
Dropout
Input: (None, 9, 2, 128) Output: (None, 9, 2, 128)
Flatten
Input: (None, 9, 2, 128) Output: (None, 2384)
Dense
Input: (None, 2384) Output: (None, 256)
Dropout
Input: (None, 256) Output: (None, 256)
Dense
Input: (None, 256) Output: (None, 512)
Dropout
Input: (None, 512) Output: (None, 512)
Dense
Input: (None, 512) Output: (None, 10)

Fig. 2. CNN Architecture used for training

3) Optimization: Adam[15] is one of the best optimizers because it trains neural networks more efficiently in less time and requires fewer parameters to tune. For sparse data, you can use dynamic learning rate. This is a combination of the gradient descent and the momentum and RMSP algorithms. This uses a gradient.

a] Momentum: In the gradient descent algorithm the convergence toward the minima of the curve is aimed by using an 'exponentially weighted average' of that gradient.

b] Root Mean Square Propagation (RMSP): It is a concept of 'exponential moving average' which aims to improve the AdaGrad. It is an adaptive learning algorithm.

Here, the above table(I & II) demonstrates various considerations of epochs and batch size, and what is the value of their accuracy and loss value. If we increase the batch size and epochs then training accuracy will increase and loss will decrease while the same for the validation phase and it's vice versa.

In general, too many epochs and batch size may cause a model to overfit the training data.

4) Denoising: In python, Noisereducer[16] (Fig. 3) is a module for denoising that helps to reduce noise in time-domain signals like in our case it is speech. It reduces noise by using the phenomena called spectrogram of a signal which is a graphical representation of time, amplitude, and frequency, and for each frequency band in the signal, it tries to fix a noise threshold which is taken as a reference to having a mask, i.e. frequency below that frequency passes.

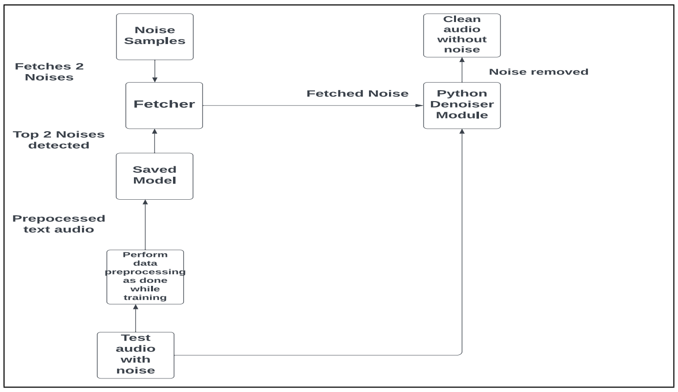

Fig. 3. DeNoiser Module Working

III. PROBLEM STATEMENT & THE PROPOSED SOLUTION

CNN is a great neural network, to train models for certain input to output. Though CNN lags behind when the inputs are not independent of each other when inputs have some relation among them. In the case of audio, it is context. When CNN is applied to these cases, it leads to a loss of information. Hence, CNN was unable to solve our problem. Hence a better way is needed.
The method should be such that it should consider the dependence of input among all other inputs. They should not be considered independent.

This problem can be solved by Recurrent Neural Network (RNN), which is a type of neural network that uses sequential data or time series data. Here, the output from the previous step is fed as input to the current step. Though, RNN is found not to store long-term dependencies and doesn't store memory for a long time, which might affect the results. Hence LSTM (Long Short Term Memory networks) was introduced to solve problems faced due to the use of RNN. It is capable of processing the entire sequence of data, apart from single data points. Information can be added to or removed from the cell state in LSTM and is regulated by gates.

So important to understand how is LSTM used to denoise. Initially, noisy audio samples are broken down into small segments. For example, if the signal is 10 seconds in duration hence break it down into each 0.1-second segment. STFT is applied on these segments to also retain frequency information. Then for each segment, data is passed to the LSTM block which has 3 gates, firstly input gate allows information to enter into the model, this information is of each segment which is sequentially given to the LSTM block as they have time relation among them. Some information is needed to be added into memory to pass on to the next LSTM block done by the cell gate and other is forgotten by forget gate.[17] Each LSTM block produces a segment of information corresponding to the input audio segment, which at last is overlapped and combined to produce the final output.

The model (Fig. 4) comprising of LSTM which can be used further to denoise is:-

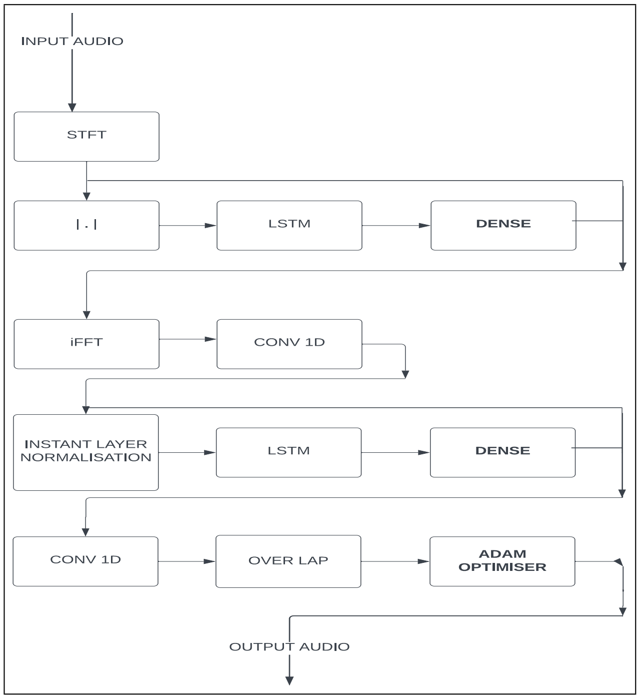

Fig. 4. Architecture of LSTM Model

where each step is described as below:-

1) STFT (Short Time Fourier Transform): The signal coming into the model is a time domain signal

Model : "functional$_1$"		
Layer(Type)	**Output Shape**	**Connected to**
Input$_1$ ($InputLayer$)	[(None, None)]	
lamba (Lamda)	[(None, None,257)]	input$_1$ [0] [0]
lstm (LSTM)	(None,None,128)	lambda[0] [0]
dropout (Dropout)	(None,None,128)	lstm[0] [0]
lstm$_1$ ($LSTM$)	(None,None,128)	dropout[0] [0]
dense (Dense)	(None,None,257)	lstm$_1$ [0] [0]
activation (Activation)	(None,None,257)	dense[0] [0]
multiply (Multiply)	(None,None,257)	lambda[0] [0] activation[0] [0]
lambda$_1$ ($Lambda$)	(None,None,512)	multiply[0] [0] lambda[0] [1]
conv1d (Conv1D)	(None,None,256)	lambda$_1$ [0] [0]
instant Layer Normalization	(None,None,256)	conv1d[0] [0]
lstm$_2$ ($LSTM$)	(None,None,128)	instant layer normalization[0] [0]
dropout$_1$ ($Dropout$)	(None,None,128)	lstm$_2$ [0] [0]
lstm$_3$ ($LSTM$)	(None,None,128)	dropout$_1$ [0] [0]
dense$_1$ ($Dense$)	(None,None,256)	lstm$_3$ [0] [0]
activation$_1$ ($Activation$)	(None,None,256)	dense$_1$ [0] [0]
multiply$_1$ ($Multiply$)	(None,None,256)	conv1d[0] [0] activation$_1$ [0] [0]
conv1d$_1$ ($Conv1D$)	(None,None,512)	multiply$_1$ [0] [0]
lambda$_2$ ($Lambda$)	(None,None)	Conv1d$_1$ [0] [0]

TABLE III

MODEL SUMMARY

to have information on frequency too. Hence STFT is used which determines the frequency and phase content of a local section of a signal as it changes over time.

2) Instant Layer Normalisation: There was a need to bring the value from a similar scale is necessary for the network to behave much better, which is called normalization. An instance normalization layer normalizes each channel for each observation independently. Here, knowledge of mean and variance is being used. In instant layer normalization, when the filter is convolved over a batch of m inputs the feature map is produced. For a batch of K inputs, the resultant feature maps are K. Instance normalization processes each of the K maps independently On each feature map, this is performed independently. Now, for each map, the mean is calculated using equation 1.

$$mean = \frac{1}{\text{height } \times \text{ width}} \sum_{r=1}^{\text{height}} \sum_{c=1}^{\text{width}} map[r,c]$$

(- Equation 1)

This is further used in equation 2 to calculate variance of each feature map.

$$var = \frac{1}{\text{height } \times \text{ width}} \sum_{r=1}^{\text{height}} \sum_{c=1}^{\text{width}} (map[r,c] - \text{mean })^2$$

(- Equation 2)

The final resultant normalized value is decided for each map cell by equation 3 which is scaled and shifted using parameter alpha and beta as shown in equation 4.

$$\text{map}[r,c]' = \frac{map[r,c] - mean}{\sqrt{var + \epsilon}}$$

(- Equation 3)

$$\text{map}[r,c]'' = \gamma\, \text{map}[r,c]' + \beta$$

(- Equation 4)

3) Conv 1D: Conv1D is a layer that convolves over the input and presents the 1D output. It uses a kernel to do this, which can be of any dimension. Dot-product of filters with an array of inputs is taken to produce an output vector. The result of each operation is a single value that adds to the output vector.

While implementing of above model, LSTM is used which represents a one-to-one mapping between noisy to clean audio while training on the dataset. The summary of the model on implementation is shown in the above Table III.

IV. Results

Denoising remained an area to research a lot for a long time. This Paper has shown different ways to enhance speech by removing noise. The observations for all those methods were done and tried to find a better method. Judging and comparing the results of each method over different noisy audios were majorly done by realizing(listening) the clean audio (output audio) and visualizing the graph of the output audio.

Here, there are three different methods used to denoise the unwanted noise namely WPT Algorithm, Convolutional Neural Network Architecture, and Long Short Term Memory Architecture. Below these are the 4 graphs which were shown in Fig 5. and Fig 6. named a) Input b) WPT Algo Output c) CNN Output d) LSTM Output
These are the outputs after applying the methods which were above mentioned. Here Y-axis represents the amplitude of the audio signal and the X-Axis represents the time of the audio signal.
It depicts the differences between the above-mentioned algorithm and gives an inference that LSTM network output shows the best output among all. For hearing the audio samples as shown in the figure, the link corresponding to each method is provided as a caption of the graph and for graphical representation as a graph hyperlink. The graph of noisy audio and its clean audio corresponding to each method are also shown by using figures. Two of them are shown in the paper.

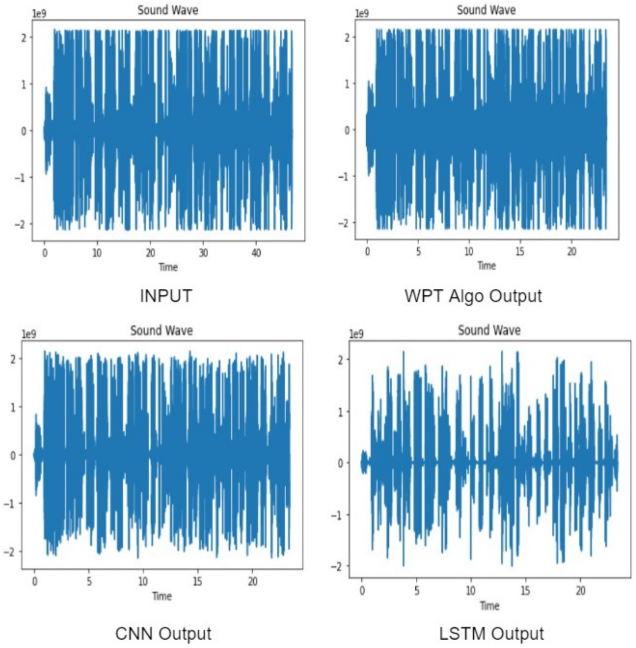

Fig 5. RESULT OF AUDIO LINKS FOR SAMPLE 1

The First Input Signal (Fig. 5) contains various types of unwanted noises with the speaker's audio that's why the graph is busier. Then unwanted noises are suppressed at each method. But the LSTM method provides the best output among all.

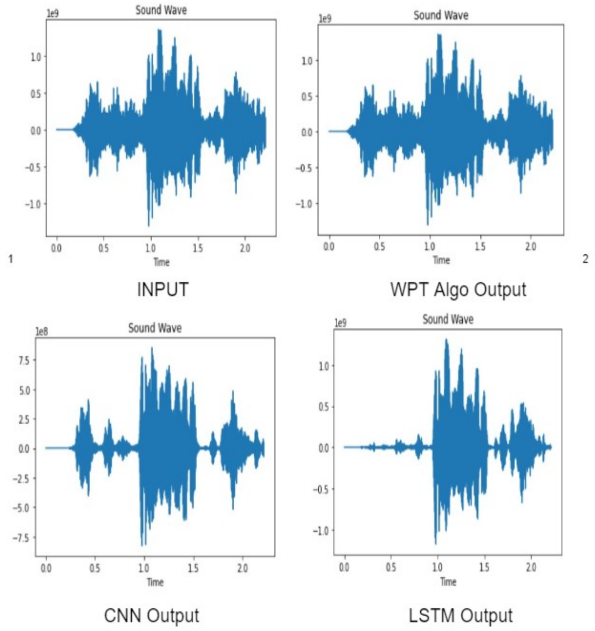

Fig 6. RESULT OF AUDIO LINKS FOR SAMPLE 2

The second input signal (Fig. 6) contains engine noise with the speaker.

Every time we progressively transition from WPT TO LSTM, the graph is seen to get clearer. Here also the LSTM method provides the best output among all.
Like these audios whose results are demonstrated above, the observation was taken for more audio too. Those graphs and noisy with clean by each method were stored in the database storage. The link to that storage is **IMPORTANT LINK**

V. Conclusion and Future Scope

This paper presented different techniques to enhance the speech. The drawbacks of WPT and CNN were discussed. It was found that these were not able to convey information after cleaning efficiently. This could be due to either noise could not be removed from input noisy audio or in order to remove noise the original signal got distorted. In both cases, the results(as presented above) were unsatisfactory, Hence, keeping in view these shortcomings, the LSTM technique was introduced. LSTM performed well because it has the ability to decide which information to keep in memory and which to forget. Results also concluded that it performed very well.

It seems that researchers still have to go a long way as various unmet problems through current techniques of denoising the audio can be observed. Currently, when audio is processed for noise suppression, some useful information might get lost. It might be possible that in order to remove noise, the original signals got distorted. The future will give the possibility to introduce better techniques to retain much information as possible. Another scope for discovery might be to use a visual of a person(could be lip-sync) to know his audio. This would allow a model to know what he/she is speaking, so the rest of the sound can be suppressed. The main challenge here again would be to retain max possible information. Hence, efficiency would surely be deciding point as to which algorithms would be adopted by the industry to meet the requirement of enhancing the audio.

References

[1] Devyani S. Kulkarni , A Review of Speech Signal Enhancement Techniques: https://www.researchgate.net/publication/301568911_A_Review_of_Speech_Signal_Enhancement_Techniques.

[2] En.wikipedia.org. Noise reduction. Available: https://en.wikipedia.org/wiki/Noise_reduction.

[3] Apoorva Athaley1 , Papiya Dutta2 , A Survey on Audio Noise Removal Techniques Available: https://www.ijarcce.com/upload/2016/si/SITES-16/IJARCCE-SITES%206.pdf.

[4] N. Alamdari, Student Member, IEEE, A. Azarang, Member, IEEE, N. Kehtarnavaz, Fellow, IEEE, Improving Deep Speech Denoising by Noisy2Noisy Signal Mapping Available: https://arxiv.org/ftp/arxiv/papers/1904/1904.12069.pdf.

[5] David Dalmazzo and Rafael Ramirez , A Review on Feature Extraction and Noise Reduction Technique Available: https://www.researchgate.net/publication/345671783_Mel-spectrogram_Analysis_to_Identify_Patterns_in_Musical_Gestures_a_Deep_Learning_Approach.

[6] CH-Olivan; IZ Pinilla, Timbre Analysis in Polyphonic Automatic Music Transcription: https://www.mdpi.com/2079-9292/10/7/810

[7] Pritish Chandna, Marius Miron, Jordi Janer, and Emilia G´omez , Paper https://www.researchgate.net/publication/313732034_Monoaural_Audio_Source_Separation_Using_Deep_Convolutional_Neural_Networks. [Accessed: 07- Jan-2018].

[8] Boyang Zhang Jared Leitner Sam Thornton, Mel-spectrogram Analysis Available: http://noiselab.ucsd.edu/ECE228_2019/Reports/Report38.pdf. .

[9] Feng Rong , Audio Classification Method Based on Machine Learning Available: https://www.researchgate.net/publication/319971531_Audio_Classification_Method_Based_on_Machine_Learning.

[10] Short-Time Fourier Transform: https://www.analyticsvidhya.com/blog/2022/03/audio-denoiser-a-speech-enhancement/deep-learning-model/.

[11] Croman and Cens: https://librosa.org/doc/main/generated/librosa.feature.chroma_cens.html

[12] M. A. Ali; P. M. Shemi, Wavelet based algorithm for audio denoising: https://ieeexplore.ieee.org/document/7455802.

[13] Authors: A Yashwanth, K Sai Shashanth, A Sangeetha, CNN : https://www.ijraset.com/research-paper/audio-enhancement-and-denoising-using-online-

[14] MATLAB denoising : https://in.mathworks.com/help/deeplearning/ug/denoise-speech-using-deep-learning-networks.html.

[15] Noisereduce Module : https://colab.research.google.com/github/timsainb/noisereduce/blob/master/notebooks/1.0-test-noise-reduction.ipynb#scrollTo=_vDSOQWpnjf7.

[16] Manmohan Dogra, Saumya Borwankar and Jayashree Domala, Noise removal from Audio using CNN and Denoiser : https://immohann.github.io/Portfolio/Denoiser.pdf.

[17] Sebastian Bock, Josef Goppold, Martin Weiß, An improvement of the convergence proof of the ADAM-Optimizer : https://arxiv.org/abs/1804.10587.