iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 10 **Issue:** VI **Month of publication:** June 2022

**DOI:** https://doi.org/10.22214/ijraset.2022.44061

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 10 Issue VI June 2022- Available at www.ijraset.com*

# Audio Enhancement and Denoising using Online Non-Negative Matrix Factorization and Deep Learning

A Yashwanth[1], K Sai Shashanth[2], A Sangeetha[3]
[1, 2]*Student,* [3]*Assistant Professor: Department of Computer Science and Engineering, Chaitanya Bharathi Institute of Technology(A), Gandipet, Hyderabad-75, Telangana, India*

***Abstract:*** ***For many years, reducing noise in a noisy speech recording has been a difficult task with numerous applications. This gives scope to use better techniques to enhance the audio and speech and to reduce the noise in the audio. One such technique is Online Non-Negative Matrix Factorization (ONMF). ONMF noise reduction approach primarily generates a noiseless audio signal from an audio sample that has been contaminated by additive noise. Previously many approaches were based on non-negative matrix factorization to spectrogram measurements. Non-negative Matrix Factorization (NMF) is a standard tool for audio source separation. One major disadvantage of applying NMF on datasets that are large is the time complexity. In this work, we proposed using Online Non-Negative Matrix Factorization. The data can be taken as any speech or music. This method uses less memory than regular non-negative matrix factorization, and it could be used for real-time denoising. This ONMF algorithm is more efficient in memory and time complexity for updates in the dictionary. We have shown that the ONMF method is faster and more efficient for small audio signals on audio simulations. We also implemented this using the Deep Learning approach for comparative study with the Online Non-Negative Matrix Factorization.***
***Keywords: Noise Reduction, Additive Noise, Non-Negative Matrix Factorization, Deep Learning.***

## I. INTRODUCTION

Generally audio contains both necessary and unnecessary audio segments in it. When audio is recorded such as speech, the speaker's required voice is recorded and also the noise of that environment is added to the audio file. This noise in the recording cause problems for the person who is listening as the speech can not be heard properly. These noises must be reduced or removed so that we get a clear speech. Removal of noises from the speech without affecting the quality of speech is known as Audio Denoising. It is also known as Speech Enhancement. It enhances the quality of speech.

Audio denoising is recently done using NMF. In this method, factorization of a non-negative data matrix is done as a product of a dictionary matrix and code matrix. While using this method, we require the entire matrix to be loaded first. This method may not be useful in applications that contain large datasets or when the input data is given in a streaming fashion.

This paper introduces Online Non-Negative Matrix Factorization (ONMF). This algorithm was developed for streaming data or when the dataset is very large to store locally. This algorithm is used to load only some part of a dataset at a time. This method uses less memory than the traditional NMF method and performs better in real-time denoising.

In this paper, we also implemented audio denoising using the Deep Learning approach for comparative study. The results of these three algorithms are compared to suggest a better approach. Compared to the previously implemented algorithms, ONMF and Deep Learning approaches are more useful for audio denoising and speech enhancement.

Therest of the paper contains the following sections. The algorithms used for this paper are explained in section II. Results and discussions are presented in section III. Conclusions and future scope are given in section IV.

## II. RESEARCH METHODOLOGY

### *A. Non-Negative Matrix Factorization*

NMF understands dictionaries for the real signal and noise from a recording that is noiseless and a pure-noise recording that is thought to be structurally similar to the signal of interest.

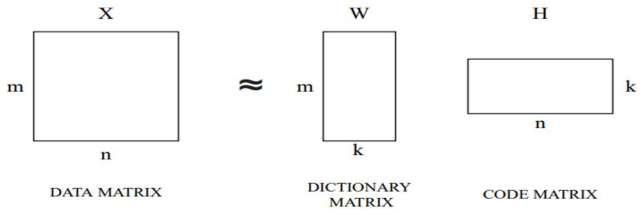



Figure 1: Matrix Factorization

In Non-negative matrix factorization (NMF), a data matrix X is factorized. It is factorized into dictionary matrix W and code matrix H. It checks that all three matrices have elements that are non-negative. Because of this non-negativity, the resulting matrices are easier to inspect.

**Algorithm 1:** NMF via Multiplicative Updates
**Input:** Data matrix X; initialization $W_0$,$H_0$
**Output:** $W_K$, $H_K$, s.t. $X \approx W_0 H_0$
**for** $k = 1,\ldots,K$ **do**
  Update Code Matrix:
  $$(H_k)_{i,j} = (H_{k-1})_{i,j} \frac{(W_{k-1}^T X)_{i,j}}{(W_{k-1}^T W_{k-1} H_{k-1})_{i,j}}$$
  Update Dictionary Matrix:
  $$(W_k)_{i,j} = (w_{k-1})_{i,j} \frac{(XW_k^T)_{i,j}}{(W_{k-1} H_k H_k^T)_{i,j}}$$
**end**

Figure 2: NMF Algorithm

*B. Online Non-Negative Matrix Factorization*

ONMF is an online matrix factorization method in which data is considered in a streaming fashion and updates the matrix factors each time. This allows factor analysis to take place in parallel with the incoming of new data samples. Unlike traditional NMF algorithms, which require the complete data matrix to be stored in memory, our ONMF method works with a single data point or a group of data points at a time. As the loading of entire data is not required in ONMF, it is more memory efficient than NMF.

Dictionary learning is a machine learning and signal processing subfield that aims to find a frame (called a dictionary) in which some training set can be represented sparsely. The better the dictionary, the more sparse the representation. The resulting dictionary is typically a dense matrix, and its manipulation can be computationally expensive both during the learning stage and later in the dictionary's usage, for tasks such as sparse coding. In ONMF, one dictionary is learned first, and then the dictionary is given training data. The data is trained using this specific dictionary. ONMF learns a dictionary better suited to representing musical chords than NMF, that does not use the time-frequency interpretation of the spectrogram, by sampling batches from this time series. Accessing the entire input data or at least a large enough training dataset is not always possible because the input data may be too large to fit into memory. This problem is solved by iteratively updating the model as new data points x become available. ONMF exhibits this type of dictionary learning. This method allows us to update the dictionary as new sparse representation learning data becomes available, reducing the amount of memory required to store the massive dataset.

**Algorithm 2:** ONMF
**Input:** Data matrix X; initialization $W_0$,
$A_0$, $B_0$= 0, regularized parameter $\alpha > 0$
**Output:** $W_T$, $H_T$, s.t. $X \approx W_0 H_0$
**for** $t = 1,\ldots,T$ **do**
  Update Sparse Code Matrix:
  $$H_t = arg\ min_{H \geq 0} \|X_t - W_{t-1} H\|_F^2 + \alpha \|H\|_1$$
  Aggregate Past Information:
  $$A_t = \frac{1}{t}((t-1)A_{t-1} + H_t H_t^T$$
  $$B_t = \frac{1}{t}((t-1)B_{t-1} + H_t H_t^T$$
  Update Dictionary Matrix:
  $$W_t = arg\ min_{W \geq 0} \frac{1}{2} \mathrm{Tr}(W A_t W_t^T) - \mathrm{Tr}(B_t W)$$
**end**

Figure 3: ONMF Algorithm

### *C. Convolutional Neural Networks*

Deep learning algorithms produce similar results to humans and are constantly analyzing data with a predefined logical structure. A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm that can take in input data, assign importance to various objects in an image or audio, and distinguish between them. These algorithms are used to denoise audio files.

The layers in CNN contain the following

*1)* Zero Padding
*2)* Conv2D
*3)* Activation
*4)* Batch Normalization

The input data to CNN has the shape (batch size, height, width, depth). The first dimension here represents the batch size of the audio sample, and the other three dimensions represent the sample's dimensions. Height, width, and depth are the three dimensions. The CNN output is a 4D array as well. The batch size is the first dimension, and it is the same as the input batch size. The other three dimensions of the output may vary depending on the filter, kernel size, and padding values we use.

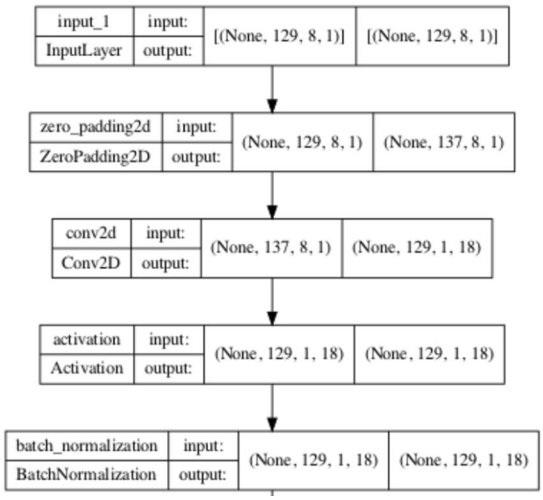



Figure 4: CNN architecture layers

To perform time-frequency analysis, we perform the Short-Time Fourier Transform method. A sequence of Fourier transforms of a windowed signal is referred to as a short-time Fourier transform (STFT). When the frequency components of a signal vary over time, STFT provides time-localized frequency information. The STFT enables us to conduct time-frequency analysis. It is used to develop representations that record both the signal's local time and frequency content. STFT is an effort to improve on the existing Fourier Transform. STFT is achieved by performing a Fourier Transform on signals and then repeatedly amplifying the signal with shifted short time windows.

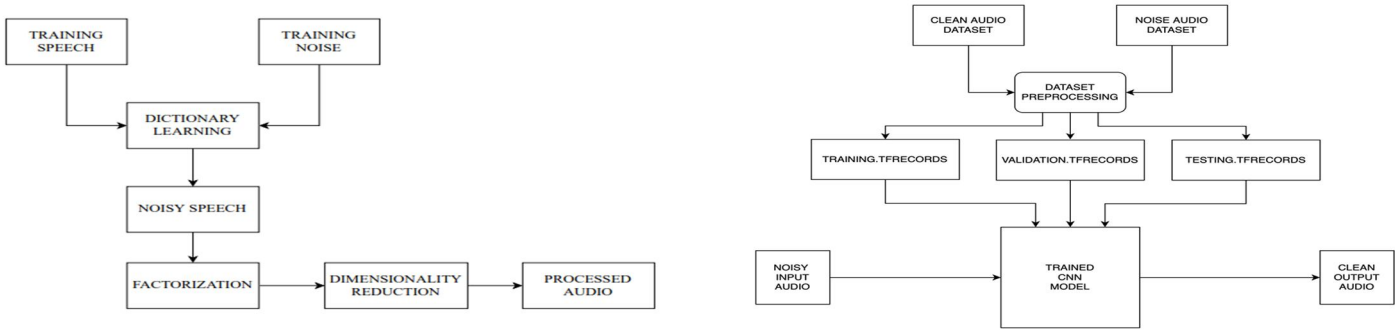



Figure 5: Flowcharts of NMF, ONMF, and CNN algorithms

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 10 Issue VI June 2022- Available at www.ijraset.com*

## III. RESULTS AND DISCUSSION

The results of the above three algorithms are discussed in this chapter. The results are compared using three performance measures.

1) *Signal-to-Distortion Ratio (SDR):* SDR is considered to be a measure that tells us how good a source sounds.
2) *Signal-to-Interference Ratio (SIR):* SIR is a performance measure that tells us the number of other sources that can be heard in an audio sample.
3) *Signal-to-Artifacts Ratio (SAR):* SAR is a measure that tells us the number of unwanted artifacts a sample has with the real source.

We used a dataset that contains audio files. The dataset is divided into two types.

*a)* Clean Audio
*b)* Noise Audio

Both datasets contain audio files of different sounds. These datasets are combined during implementation to make noisy audio. These noisy audio files are used to build the model. For training we considered 1000 audio files, for testing and validation we considered 100 audio files each. One dataset contains clean audio without any noise in it. This contains clean speech and clean audio of other sounds. Another dataset contains noise audio with noise in it. These datasets are combined to make a noisy dataset.
In NMF and ONMF, the audio file is given as input. The audio file is converted to dictionaries, from which the training and testing process is done. The audio cannot be directly processed, so it is converted into different transforms. Fast Fourier Transform and Short-Time Fourier Transform are applied to convert audio. The samples of clean and noise audio are converted to spectrograms. From these spectrograms, dictionaries are learned using the NMF approach by minimizing the loss function. The dictionaries of clean and noisy samples are combined to form a coding matrix, this matrix is used future to decompose the noisy audio into clean and noise samples. In ONMF, one dictionary is learned first and training data is given to the dictionary. From this particular dictionary, the data is trained.
In CNN, We take Noise and clean audio. We merge those audios to form a dataset called noise dataset. Then we extract features such as Mel spectrogram, normal, frequency, and domain from Audio. And next, we train our model using these spectrograms.

### *A. Non-Negative Matrix Factorization*

The denoising process is done followed by plotting of Spectrograms of input and output signals and finding the performance using performance measures.

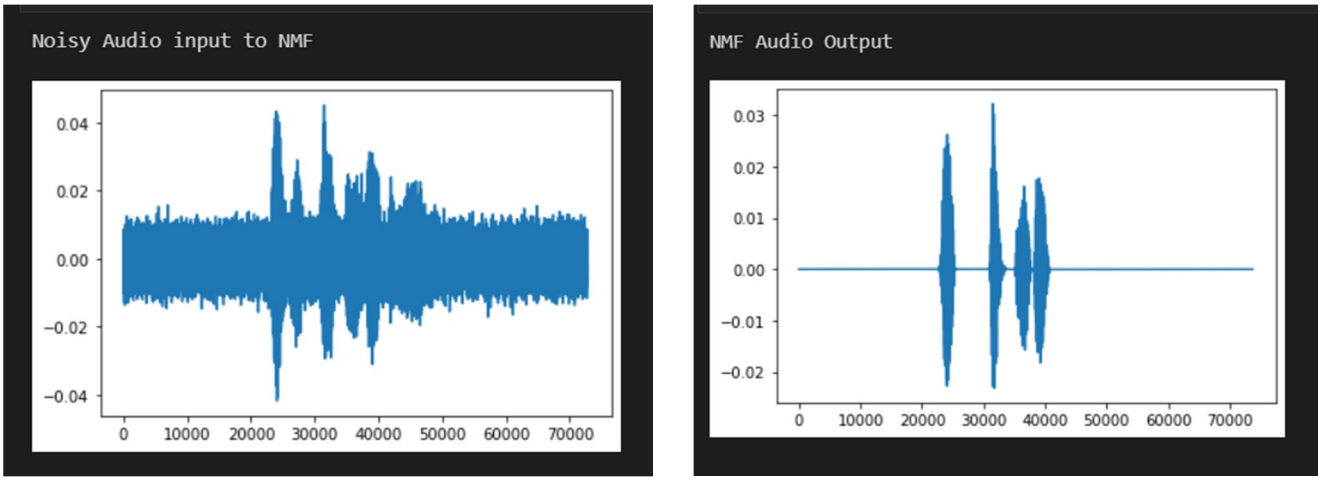



Figure 6: Spectrogram representation using NMF

Spectrograms of noisy input and output clean audio signals are displayed. By looking at the spectrograms, the noise that is removed can be identified. The amplitude of the signal is completely clear which tells that some part of the real speech is removed as noisy speech from the audio. Performance is checked by using three standard performance measures SDR, SIR, and SAR. Their values are compared between clean and processed audio signals. Histograms of these measures are plotted to differentiate.

```
SDR for Clean Signal    : 12.606410416623874
SDR for Processed Signal    : 25.20966836877477
SIR for Clean Signal    : 10.82384845948578
SIR for Processed Signal    : 22.52383093664482
SAR for Clean Signal    : 11.169013586927715
SAR for Processed Signal    : 22.54805179955927
```

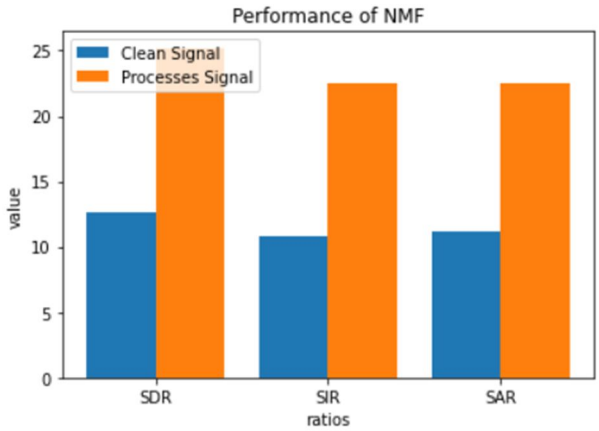



Figure 7: Performace measures of NMF

*B. Online Non-Negative Matrix Factorization*

The denoising process is done followed by plotting of Spectrograms of input and output signals and finding the performance using performance measures.

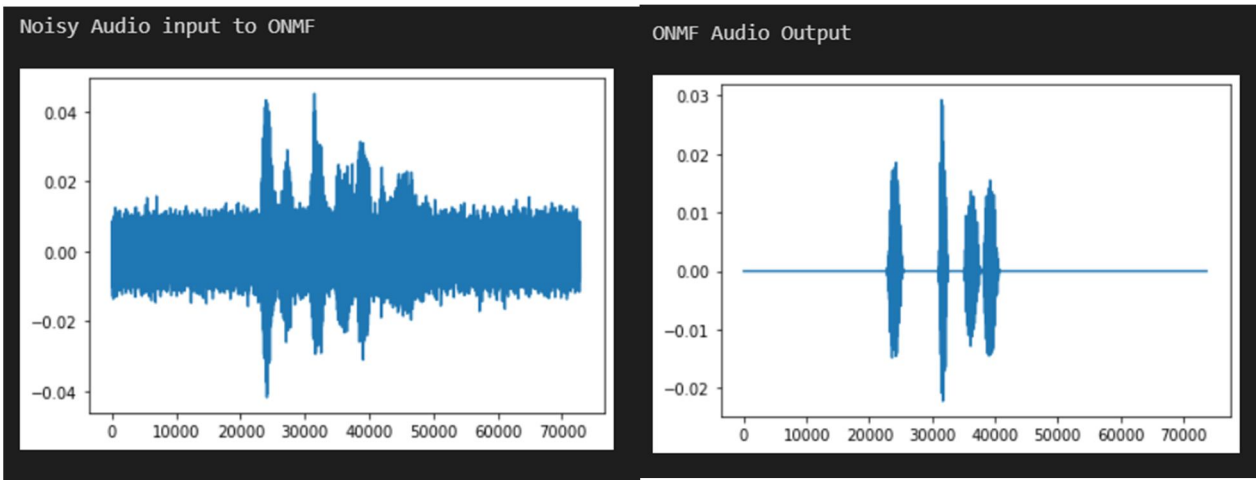



Figure 8: Spectrogram representation using ONMF

Spectrograms of noisy input and output clean audio signals are plotted. By looking at the spectrograms, the noise that is removed can be identified. The amplitude of the signal is completely clear same as in NMF which tells that some part of the real speech is removed as noisy speech from the audio after denoising.

```
SDR for Clean Signal    : 12.606410416623874
SDR for Processed Signal    : 27.85699117479541
SIR for Clean Signal    : 10.82384845948578
SIR for Processed Signal    : 25.08831219813449
SAR for Clean Signal    : 11.169013586927715
SAR for Processed Signal    : 25.10174854172897
```

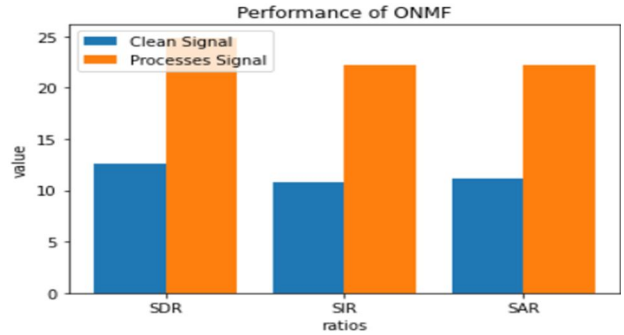
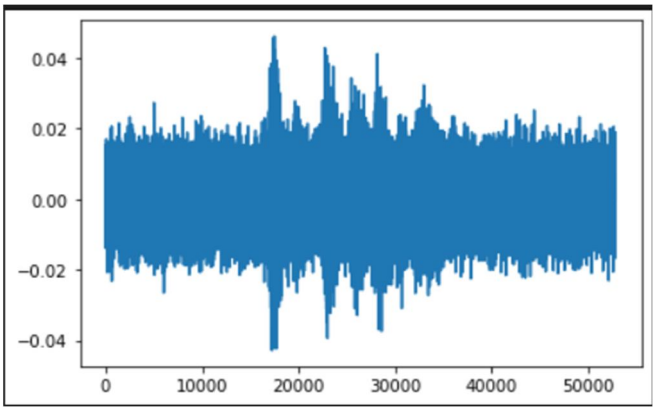



Figure 9: Performance measures of ONMF

### *C. CNN*

For input of clean and noisy audio, we considered the same inputs given to the other two algorithms as we did a comparison between the algorithms.



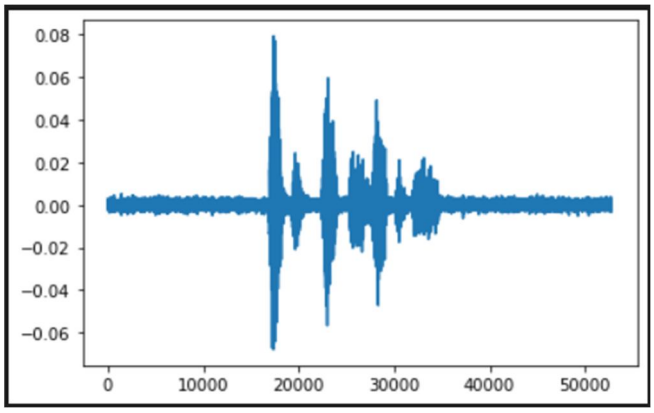




Figure 10: Spectrogram representation using CNN

```
SDR for Processed Signal    : 25.916585192283303
SIR for Processed Signal    : 23.206274658212198
SAR for Processed Signal    : 23.226981843749744
```

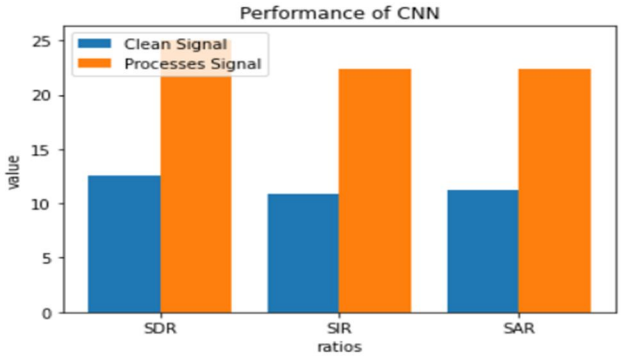



Figure 11: Performace of CNN

*1)* After implementing NMF and ONMF algorithms, the output spectrograms are almost similar. In both the algorithms the noise is completely removed and the performance is calculated using three standard performance ratios.
*2)* The performance of ONMF is comparatively better than NMF. We can find this by looking at the Histograms.
*3)* These algorithms are then compared with the deep learning-based algorithms.
*4)* While implementing, ONMF has taken more time to get the output results. This is because dictionary learning takes more time.
*5)* Both ONMF and NMF removed necessary audio segments which are not completely noisy but may contain some disturbance.
*6)* While the Deep learning-based model did not remove all the noise and gave proper audio output with necessary audio and removed noisy audio from it.

## IV. CONCLUSION

This chapter concludes the paper and discusses the future scope of implementing this.

After looking at the results, we conclude that using ONMF, even though we get better performance measures, removing complete noise sometimes may not be useful. But by using CNN, only the noise which is not necessary is removed and performed almost equal to ONMF. Time taken for generating output is also less in CNN compared to ONMF. ONMF is more memory efficient but time-consuming. Finally, we conclude that using CNN based denoising model can give us better results than ONMF and NMF.

In ONMF, during noise separation, some of the real noise is getting removed from the audio. ONMF takes more time for dictionary learning, and batch formation. In the future, one can develop an ONMF algorithm to reduce the removal of real noise from the audio. This can be done by developing an audio denoising model using more efficient deep learning models, using more layers, and running the dataset more times to reduce the loss and increase the performance of the algorithm so that we get a clear speech or audio after removing noise and enhancing the audio.

## REFERENCES

[1] Andrew Sack, Wenzhao Jiang, Michael Perlmutter, Palina Salanevich, and Deanna Needell, "On Audio enhancement via Online Non-negative Matrix Factorization", arXiv:2110.03114v1 [eess.AS], 2021.

[2] Kwang Myung Jeon, Geon Woo Lee, Nam Kyun Kim, and Hong Kook Kim, "TAU-Net: Temporal Activation U-Net Shared With Nonnegative Matrix Factorization for Speech Enhancement in Unseen Noise Environments" in IEEE/ACM Transactions on Audio, Speech, and Language Processing ( Volume: 29), 2021, pp. 3400-3414, doi: 10.1109/TASLP.2021.3067154.

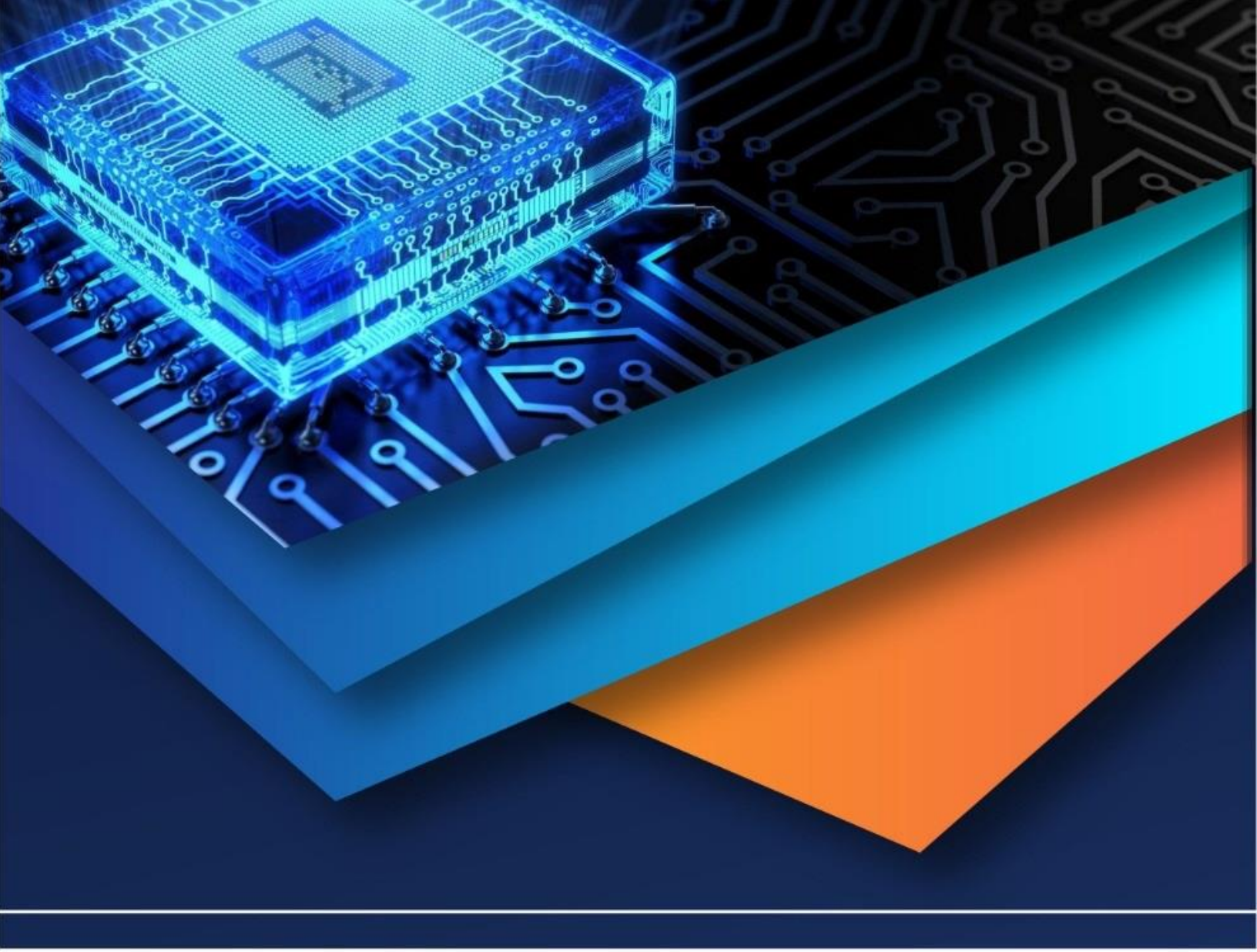
[3] Hanbaek Lyu, Georg Menz, Deanna Needell, and Christopher Strohmeier, "Applications of Online Nonnegative Matrix Factorization to Image and Time-Series Data" in Conference of 2020 Information Theory and Applications Workshop (ITA), 2020, doi: 10.1109/ITA50056.2020.9245004.

[4] Augustin Lefèvre, Francis Bach and Cédric Févotte, "Online algorithms for Nonnegative Matrix Factorization with the Itakura-Saito divergence" , 2011. hal-00602050, https://hal.archives-ouvertes.fr/hal-00602050.

[5] Hanbaek Lyu, Deanna Needell, and Laura Balzano. Online matrix factorization for markovian data and applications to network dictionary learning. Journal of Machine Learning Research, 21(251):1–49, 2020.

[6] E. Vincent, R. Gribonval, and C. Fevotte. Performance measurement in blind audio source separation. IEEE Transactions on Audio, Speech, and Language Processing, 14(4):1462–1469, 2006.

[7] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. Journal of Machine Learning Research, 11(1), 2010.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY