Sustainable Computational Engineering

Final Report

shuteng wang

July 14, 2022

Contents

1 Abstract

Accurate localization of acoustic sources using collected acoustic signal remains an important application in structual health monitoring, aeronautics, ocean science and related areas. Due to the intrinsic complexity of acoustic signal processing, it is always hard for localization with high accuracy. Although, there exist traditional methods, for example, using triangulation rules and the relative distance between at least two mics, the rough acoustic source could be spotted. However, this requires complex data processing, filtering of signals and the parameter selection is also tricky. So in this paper, we tried to use the power of artificial intelligence and test its performance on this problem. With our designed network, it achieves an accuracy of about 0.13 cm on the test localization problem.

2 Introduction

Acoustic signals contain rich information and can always help in sensing and understanding the environment around. In recent years, using mic arrays to record the phase interference pattern of acoustic signals is proven to be helpful for accurate localization purpose.[1] Due to the advancement of technology, collecting tons of acoustic signals using microphones could be easily achieved. Properly using this signal and processing it to extract informative features, however, still remains application-dependent and requires arduousness.

The common approaches for feature design in audio signal processing include fast fourier transforms(FFT), spectrograms using short-time Fourier transform(STFT)[2], wavelet transformations[3], mel-frequency coefficients[4] and some others. After processing raw signals to get the most effective features, we could further analyse the data in a model-driven or a data-driven approach.

Recently, neural networks has been more widely used for this kind of problems to help us process the hand-designed features, or exploit new representative features from the raw signals directly. Deep neural networks(DNN)[5], Convolutional neural networks(CNN)[6], Residual networks(ResNet)[7], and AcousticNet[1] etc, they all tried to use the features to find the pattern hidden inside acoustic signal and localize the acoustic source with a high accuracy. One possible problem, that has never been properly considered is the variance

of the feature weights among different mics for acoustic sources localization. Therefore, in this paper, we presented a new network that can self-adjust different mic feature weights using attention mechanism for possibly different acoustic sources.

3 Problem statement

In the work, we prepared an array of 112 microphones constituting a mic array as shown in fig 1.

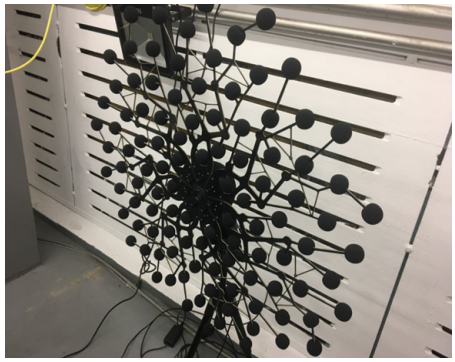

Figure 1: Mic array arrangement

We also prepared a steel plate with point grid marked on the surface. A high-density steel ball is released with fixed initial angle to hit points on that grid while the acoustic signal is being collected as shown in fig 2.

Due to the difference in the wave propagation speed on different mediums, and the relative difference on the source-to-mic distance, we could obtain a map of dense signal interference patterns. We believe this interference pattern could grant us sufficient information for signal source detection after.

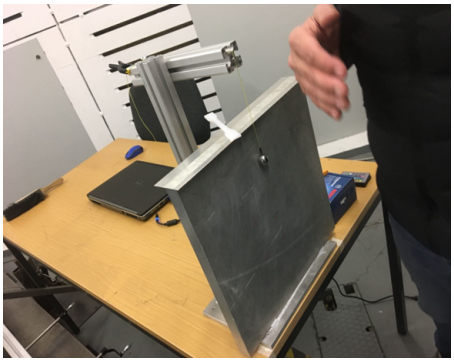

Figure 2: Steel plate with steel ball releasing

4 Method and feature design

In this part, we tried to compare different hand-designed features and use the best feature as our input for the neural network. With our raw signal including re-bounces of the initial hit, we choose to use the Python built-in package to cut the data into separate segments corresponding to different re-bounces as shown in figure 3.

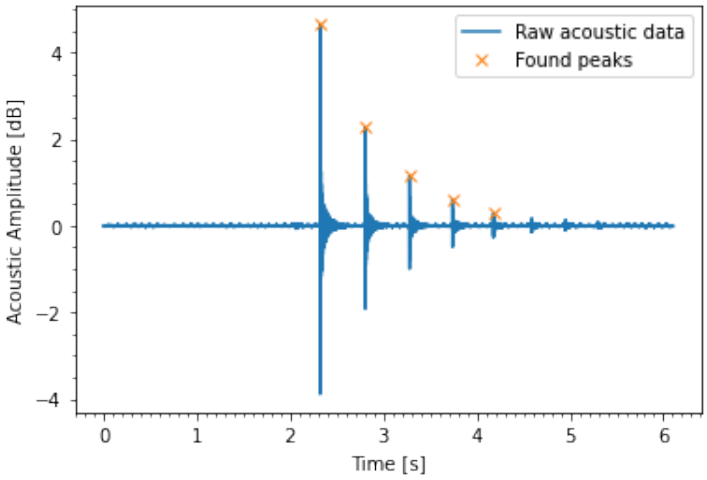

Figure 3: Raw continuous signals

Part of the raw signals collected by the mic array is shown in fig 4 to give

an impression of how the pattern looks like.

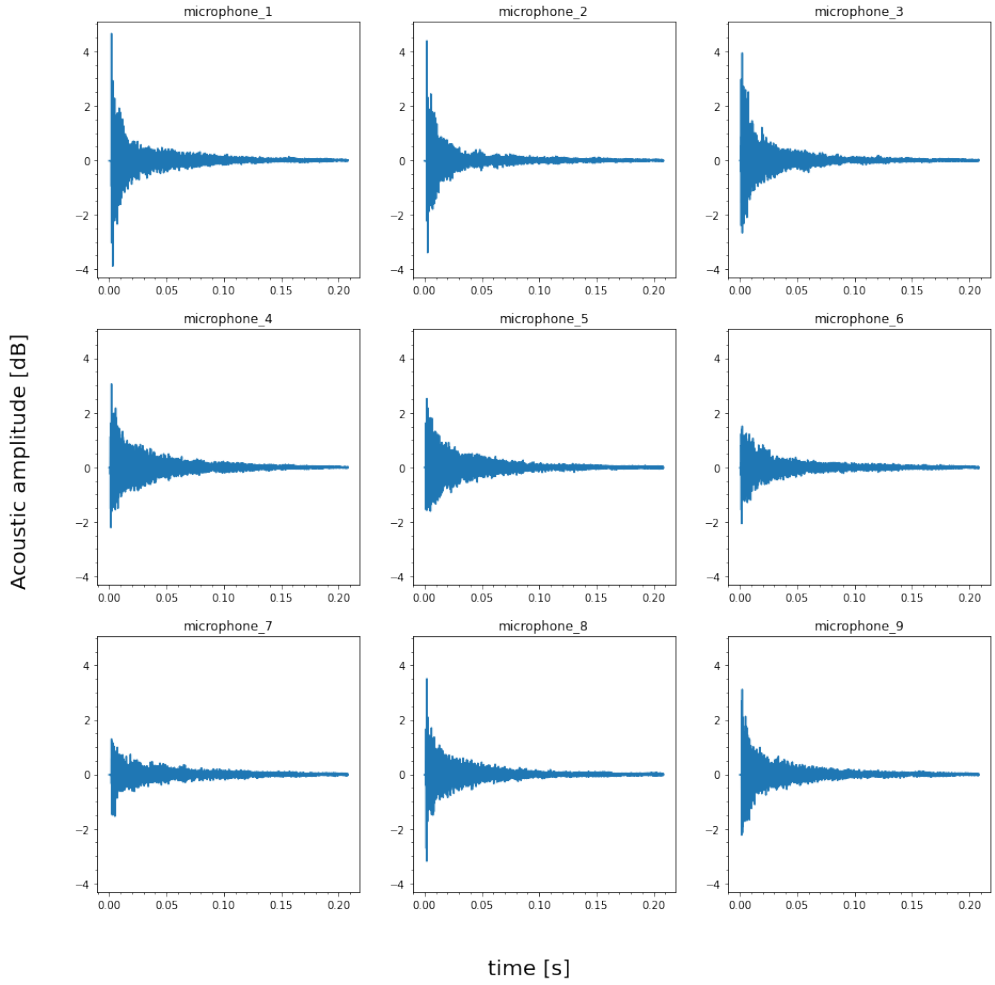

Figure 4: Signal interference pattern collected by the mic arrays

4.1 Fourier transform and its variant

As our first try, Fourier transform, as shown in fig 5, is implemented to get us an intuition on the frequency range the signal lies on.

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \tag{1}$$

From the frequency plot, it could be observed that most of the data lie below the frequency of around 11kHz.

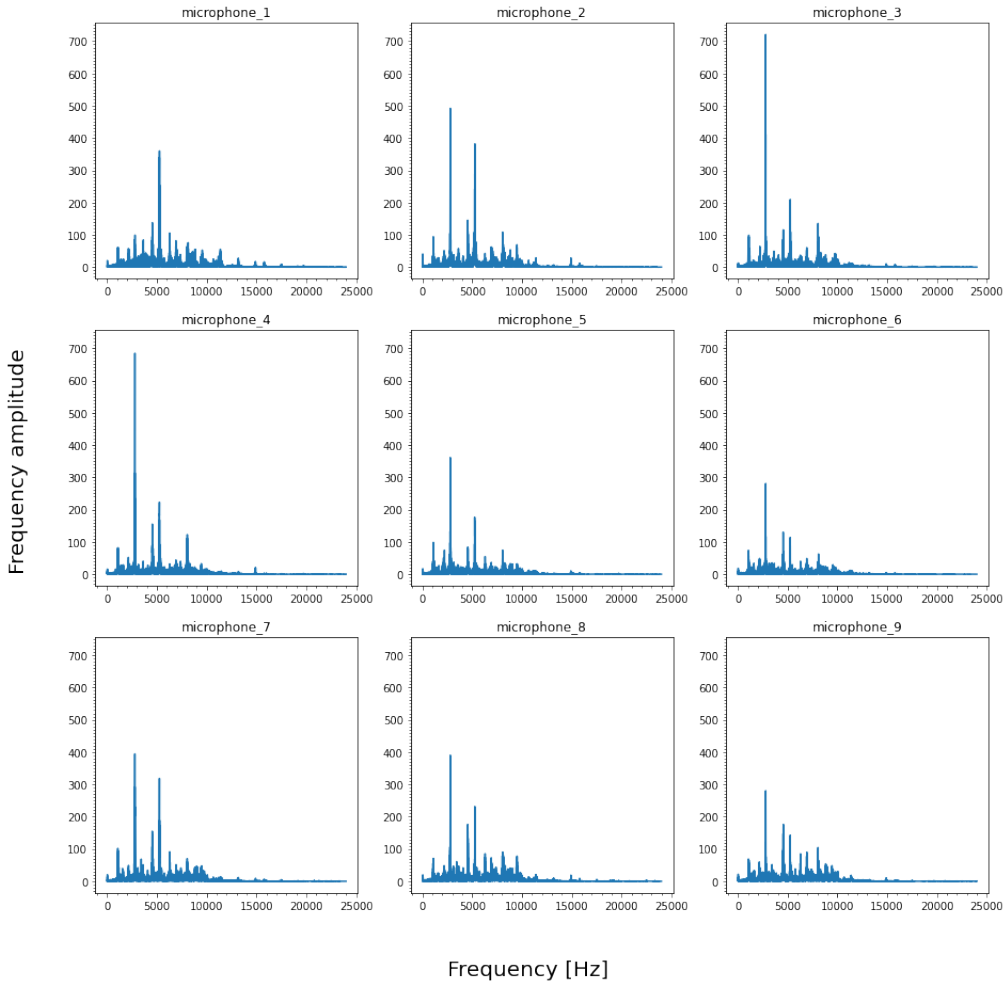

Figure 5: Fourier transform of raw signals

However, since fourier transform is a global algorithm for the time-to-frequency conversion that discards temporal information completely, it might create unnecessary noise and cause low-fidelity issues if the signal pattern is unsteady in the time domain. To get rid of this problem, we also tried one fourier transform variant, the so-called short-time Fourier transform(STFT), which creates 2D spectrograms as shown in fig 6.

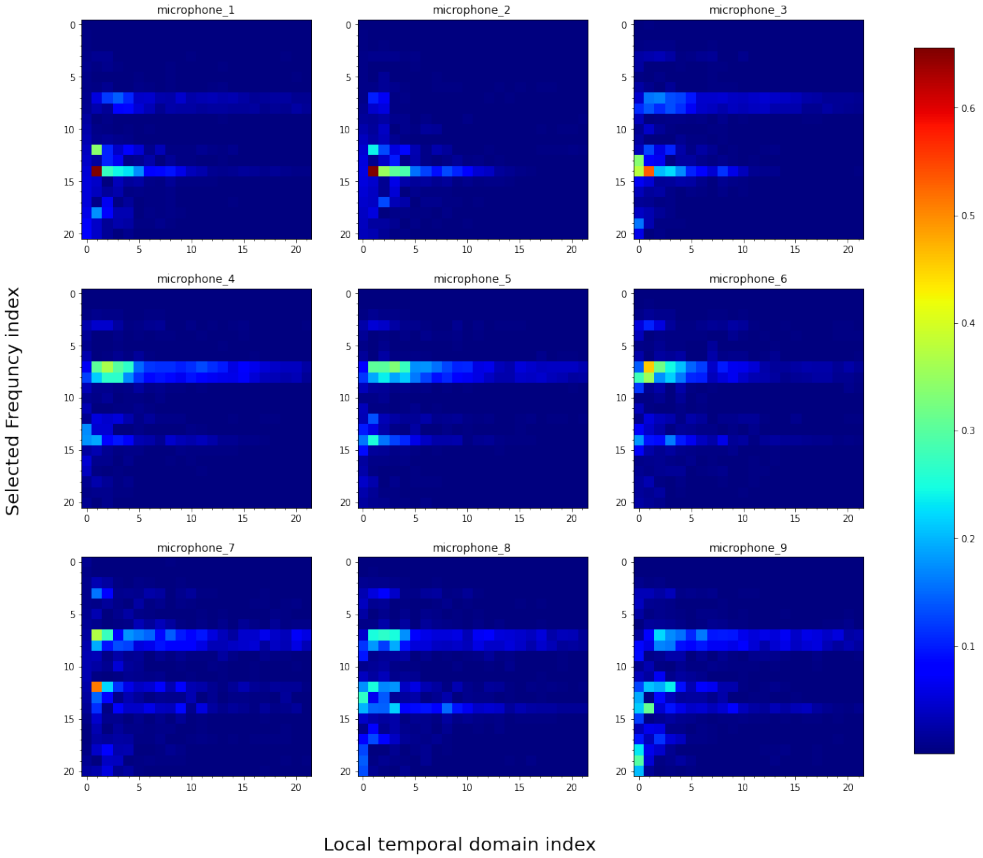

Figure 6: spectrograms of raw signals

This approach splits the entire temporal domain to several smaller time windows, and performs fourier transform on each small window respectively with details given in equation 2.

$$F(\tau, \omega) = \int_{-\infty}^{\infty} f(t) w(t - \tau) e^{-i\omega t} dt \tag{2}$$

with ω being our frequency, f(t) is our signal function, $w(t - \tau)$ being the window function and τ contains our temporal information. This way, it purposefully adds limited temporal information, depending on our choice of parameters for the domain split. However, the selection of this parameter is dominated by the so-called uncertainty principle, stating that the smaller we make the size of the window the more we will know about where a certain

frequency has occurred in the signal, but less about the frequency amplitude itself. The larger we make the size of the window the more we will know about the frequency amplitude but less about the temporal information.[8]

4.2 Wavelet transformation

Although Fourier transform and Short-time Fourier transform both perform time-to-frequency transformation, we want to introduce a new feature representation called "wavelet transformation" which performs it differently. Fourier transform uses a series of steady sine waves to approximate the original signal, which means the raw signal needs to be a combination of steady sine waves with different frequencies along the global temporal domain. One drawback, as mentioned above, is its poor performance for transformation of unsteady signals. As shown below in fig 7, fourier transform could locate all consisting frequencies for steady signals, however, there comes with small oscillations in the signal frequency-transition phase, which I would refer to as noise and fidelity loss.

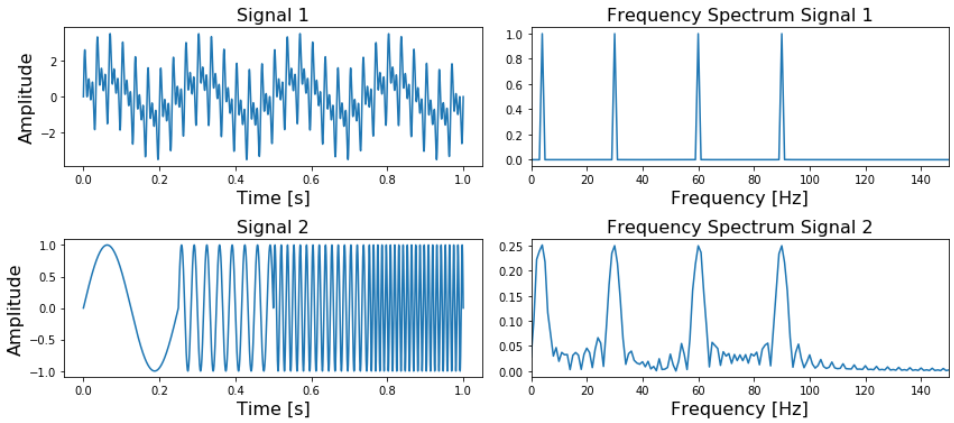

Figure 7: The signals and fft of a signal which contains four frequencies at all times (top),four different frequencies at different times (bottom)[8].

Instead of using series of sinusoidal functions for calculating frequencies, wavelet transform uses wavelets to achieve this purpose. The difference between a sinusoidal wave and a wavelet is shown in fig 8. The wavelet has two important parameters, scales and translations. The parameters are explained in detail with fig 9.

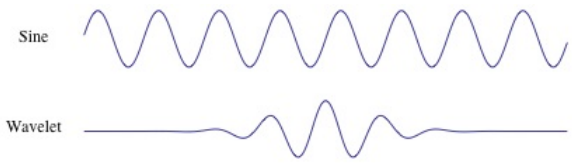

Figure 8: The difference between a sine-wave and a wavelet. The sine-wave is infinitely long and the wavelet is localized in time[8].

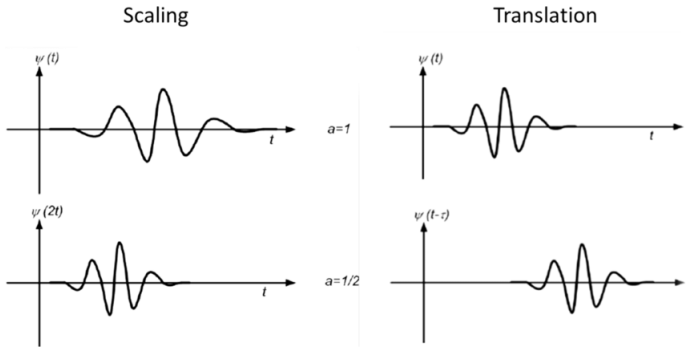

Figure 9: Scaling and translation of a wavelet[9]

The sine waves propagate along the domain without providing temporal localization, while a wavelet is indeed localized in time. Using a wavelet, we could convolve it with the raw signals and produce scaleograms as features. There are many types of wavelet, depending on the compactness and smoothness of the wavelet functions, and they can be used for different feature extraction.

Wavelets are basically functions that:

- Have finite energy, which means localization in temporal space and frequency space, thus integrable and convolution always exists at every point.
- Have zero mean in both temporal domain and spatial domain, which guarantees its integrability, existence and calculatibility of the inverse of wavelet transform.

4.2.1 Continuous wavelet transform

One category of wavelet transform is continuous wavelet transform(CWT). CWT is described by the following equation:

$$X_w(a,b) = \frac{1}{|a|^{1/2}} \int_{-\infty}^{\infty} x(t)\overline{\psi}(\frac{t-b}{a})dt \tag{3}$$

where $\psi(t)$ is the continuous wavelet function with a continuous scale factor of a and continuous translation factor of b. With CWT applied to a 1D signal, the resulting scaleogram would provide us with dynamic state-space behaviour of the raw signal. The scaleogram also suffers from the precision loss that could be explained in a more general form from the famous "Heisenberg Uncertainty principle" in quantum mechanics, stating that there always exists a trade-off of resolution between position and momentum. Here, in this case, the uncertainty theorem states that the smaller scaling, the higher resolution in temporal domain and coarser resolution in frequency domain and vice versa.

A plot called "Heisenberg box" is shown in fig 10. These plots show us the time and frequency resolutions of the raw signals, fourier transformations, STFT and the wavelet transformations respectively. The uncertainty formula that is used to describe this could be found below.

$$\Delta t \Delta f \geq \frac{1}{4\pi} \tag{4}$$

Where Δt is the temporal resolution and Δf is the frequency resolution. It could be clearly observed from the Heisenberg box that the STFT has medium resolution for both time and frequency domain. Wavelet has high temporal but compromised frequency resolution for low scaled wavelets and high frequency resolution with compromised temporal resolution for high scales wavelets. This to the largest extent, satisfies our need for accurate, high-fidelity frequency information extraction from the raw signals.

- Complex Morlet Wavelet
  Complex Morlet Wavelet(CMW) is a commonly used wavelet function as shown in 5 for analysing unsteady signals.

$$\psi(t) = \frac{1}{\sqrt{\pi B}} exp^{-\frac{t^2}{B}} exp^{i2\pi Ct} \tag{5}$$

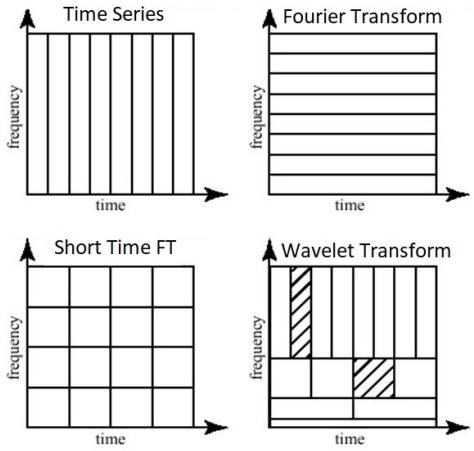

Figure 10: Heisenberg box[8]

With B being the bandwidth and C being the center frequency. The plot of the real part and the imaginary part for a complex Morlet wavelet is shown in fig 11. It can be realized that this is indeed a Gaussian modulated sinusoidal function using Euler's rule. By convolving this wavelet with the raw signal, we could obtain the specific power and phase information from the raw signal corresponding to the real part and imaginary part based on the parameters of the selected wavelet function.

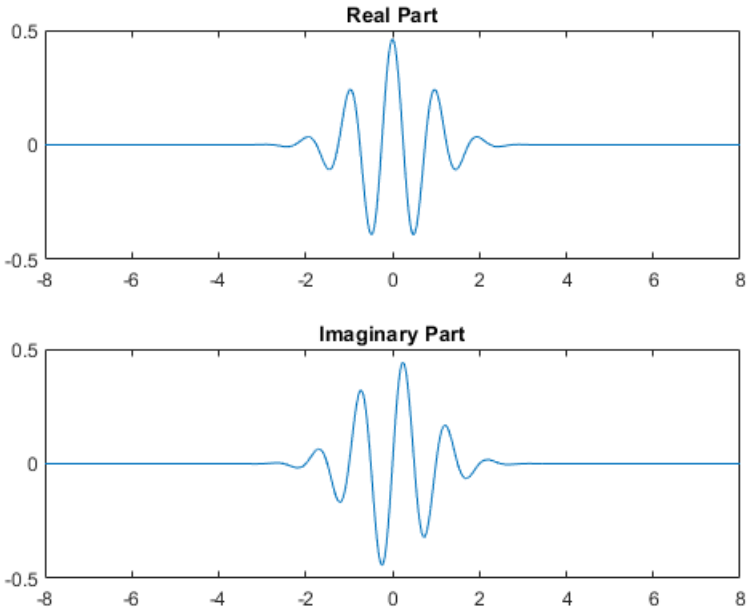

Figure 11: Complex Morlet wavelet with B = 1.5 and C = 1.0

In this work, we tried to use the convolved signal with complex Morlet wavelet as our features that we would later use as input for our neural network. Since our convolved signal is complex-valued with real and imaginary parts representing power and phase information, we calculated the absolute value of the complex signal and combined the phase information with the power information. Although this may decrease the feature distinguishability, this dramatically decreases the feature dimensionality which concerns future resources usage.

Based on previous knowledge and the information on salient useful frequencies from the Fourier transform, we obtained the scaleogram with focuses on the frequency range between 1 KHz and 11 KHz using the complex Morlet Wavelet cmor1-1.5 (bandwidth 1 and center frequency 1.5) as shown below in fig 12.

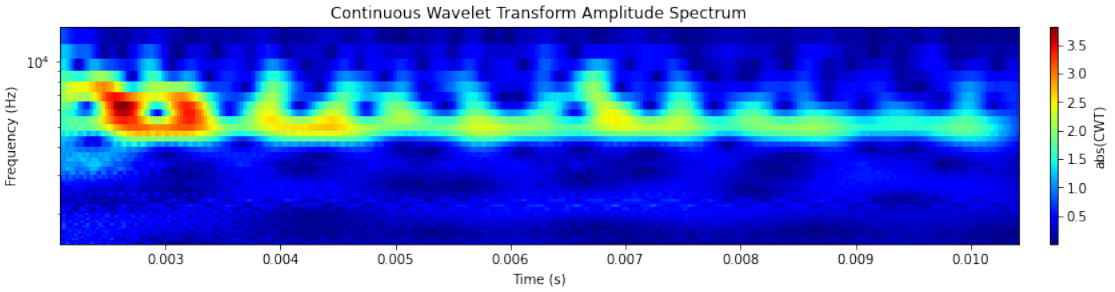

Figure 12: Scaleograms obtained with complex morlet wavelet

5 Principle component analysis

Since directly feeding the scaleograms to further processing pipelines would require a large math model and significant computational power, we choose to use the trick of principle component analysis(PCA) to further compress the data and keep reducing the dimensions of data by projecting the data on to the set of new orthogonal axes, while preserving most of data's variances. A 3D demo of PCA is given below for illustration purpose in fig 13.

With PCA applied to original data, we can find set of orthogonal linear transformations of original data such that the greatest variance is preserved on the 1st axis, second greatest variance is preserved on the 2nd axis and so on. Principal component analysis helped us to achieve this by calculating the eigen-decomposition of the covariance matrix of the data points. The eigen-vector which matches the largest eigenvalue is used as the 1st principle axis, and the one that matches the 2nd largest eigenvalue is our 2nd principle axis and so forth. By applying PCA to our scaleogram with preservation of 98.8 percent variance, we got the pca-processed scaleogram features as shown in fig 14 with a large dimensionality reduction.

6 Neural network structure

Self-attention mechanism, and its later variants firstly appear to solve the tasks in the field of natural language processing(NLP) to mimic the behaviour of human cognition to amplify the important data features while suppressing those unimportant ones[11]. Then comes with more advanced mechanism

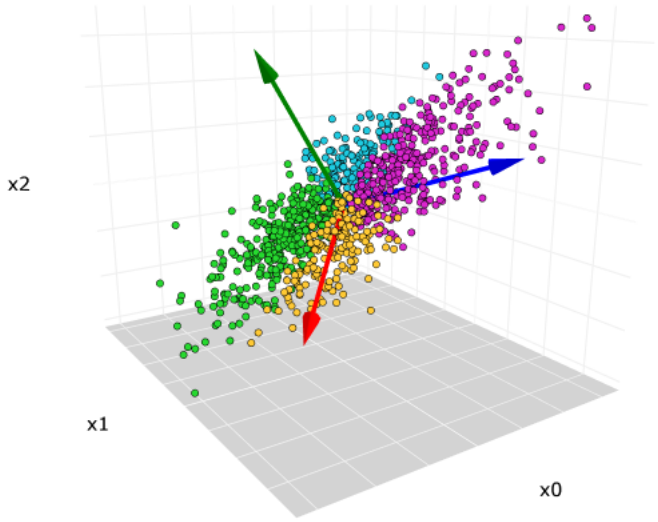

Figure 13: Principle component analysis for 3D data analysis[10]

that implements self-attention by densely projecting features to queries, keys and values[12]. By cross-multiplying the queries and keys of each feature patch, we could obtain the relative weight of each feature patch which tells us how much attention we should put on each other.

After the big success in NLP, researches also tested its performance in computer vision(CV) area and designed Vision Transformer(ViT)[13], multi-scaled Vision Transformers(mViT)[14] and improved multi-scaled Vision-Transformer(mViT2)[15] etc. Multi-head self-attention promotes the basic idea of self-attention in a more general way to include multiple channels of features, and multiple channels of keys and queries.

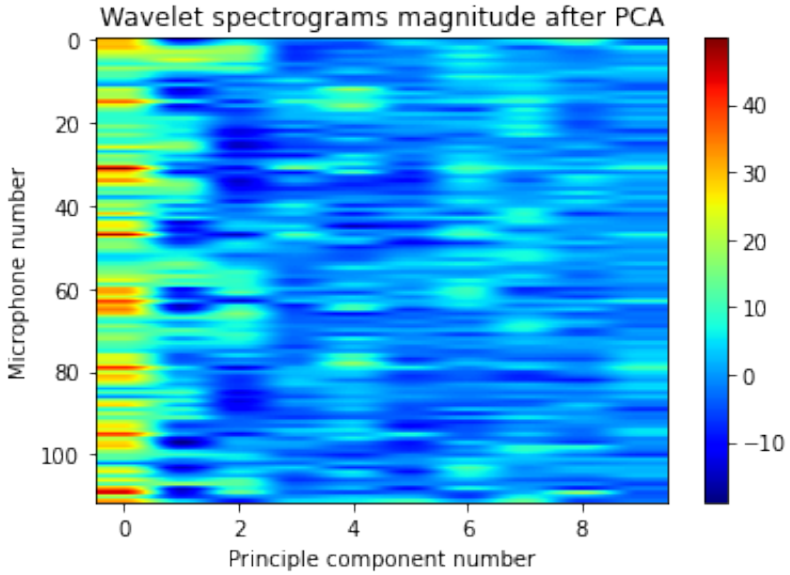

Figure 14: PCA transform of scaleogram features

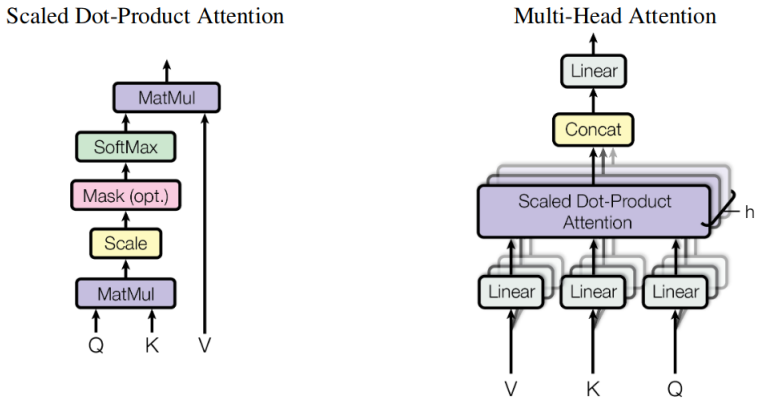

Figure 15: multi-head self-attention mechanism[12]

Our designed network is shown in fig 16. In this network, we used the idea of this self-attention mechanism in the hope that it could self-adjust the feature weights. Our input is the PCA-processed features from all the mics. We firstly performed the non-linear dense projection on all the mic features for useful information extraction, then we add dropout and normalization for regularization purpose to prevent overfitting and smoother curve of training.

To implement our attention mechanism, we referred to the idea of the ViT[13] with little modifications and tested the performance of this architecture on the acoustic data. The optimizer is set to Adam with initial learning rate of 0.001. Shuffling and early stopping are also used for easier learning and prevention of overfitting respectively. The whole task runs on Nvidia Tesla V100 SXM2 with 16GB memery in RWTH GPU compute cluster.

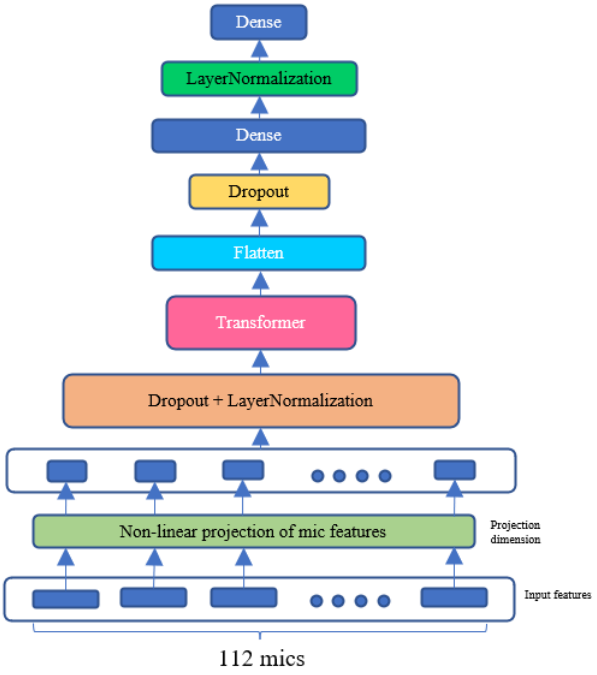

Figure 16: Artificial neural network structure

7 Visualization of the train/test data

Because of the intrinsic difficulty and properties of the regression problem, the train/test dataset is manually created in the way that the test points are always surrounded by train points to make sure the network learns the regression pattern inside the so-called "valid domain".The visualization of the train/test points are shown below in figure 17.

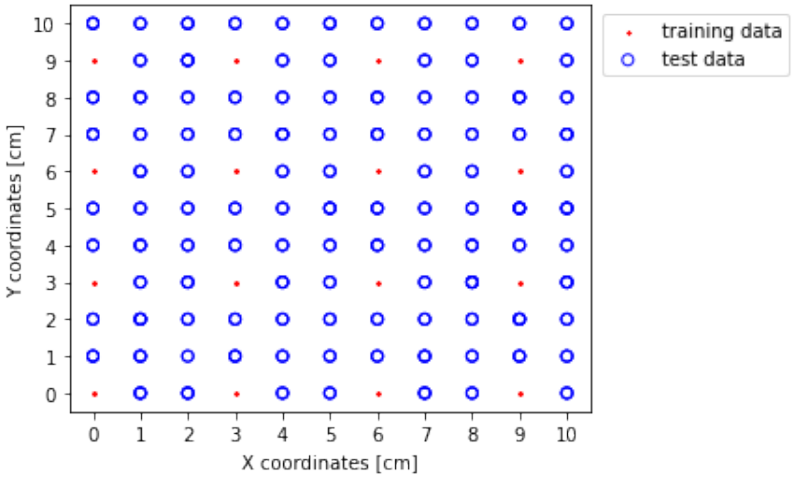

Figure 17: Train/test points visualization

8 Results visualization

With our designed network, the training loss is shown below in figure 18. As could be observed from the figure, the training curve is quite smooth and the training loss and test loss synchronize quite well which indicates that the learnt pattern fits quite well to the test data.

The final loss for our training data, using the metric of mean absolute error(MAE) reaches 0.11 cm, and for the test points, the MAE reaches 0.13 cm, which is indeed a satisfying result for acoustic localization problem. The network predictions could also be visualized below in figure 19.

The red points are the original hit locations, the blue points are the predicted training points and the green points are the predicted testing points.

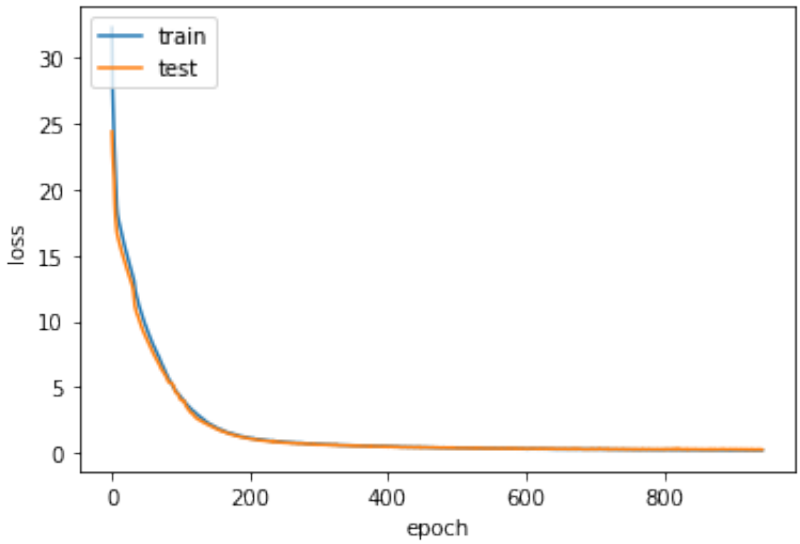

Figure 18: Training loss

It can be confidently concluded that even with the noise/imperfections existing in the experiment environment, this network does capture the pattern and generalize quite well.

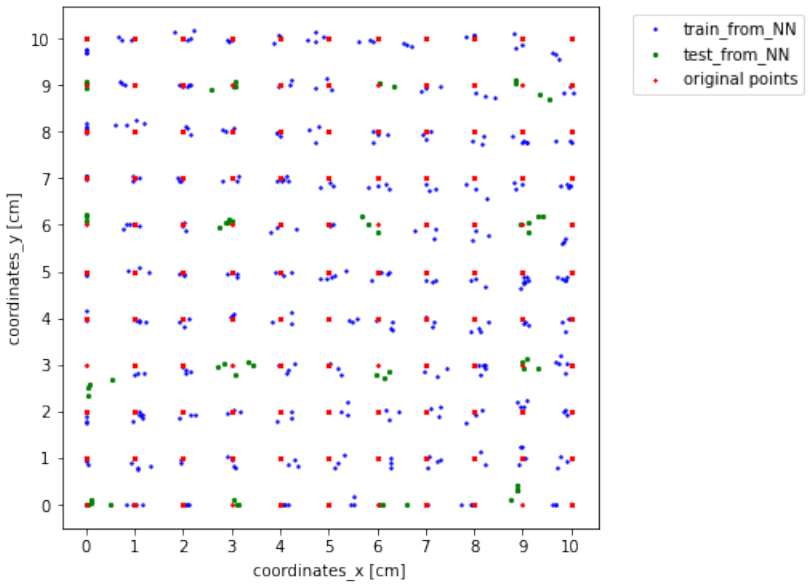

Figure 19: Neural network prediction

9 Discussion and conclusion

In this project, we presented a novel architecture that use the concept of self-attention to automatically adjust the feature weights of different mic features inside the mic array to better learn the patterns hidden inside the data. This new idea not only gives researchers a new alternative methods including the traditional ones on acoustic localization, but also provides an approach on automating feature weights in a scenario where multiple sensors are available.

References

[1] G. Zhou, H. Liang, X. Ding, Y. Huang, X. Tu, S. Abbas, Acoustic-net: A novel neural network for sound localization and quantification, arXiv preprint arXiv:2203.16988 (2022).

[2] F. Feng, Y. Ming, N. Hu, Sslnet: A network for cross-modal sound source localization in visual scenes, Neurocomputing (2022).

[3] T. Li, Z. Zhao, C. Sun, L. Cheng, X. Chen, R. Yan, R. X. Gao, Waveletkernelnet: An interpretable deep neural network for industrial intelligent diagnosis, IEEE Transactions on Systems, Man, and Cybernetics: Systems (2021).

[4] J. Sangeetha, R. Hariprasad, S. Subhiksha, Analysis of machine learning algorithms for audio event classification using mel-frequency cepstral coefficients, in: Applied Speech Processing, Elsevier, 2021, pp. 175–189.

[5] J. Yangzhou, Z. Ma, X. Huang, A deep neural network approach to acoustic source localization in a shallow water tank experiment, The Journal of the Acoustical Society of America 146 (6) (2019) 4802–4811.

[6] W. Liu, Y. Yang, M. Xu, L. Lü, Z. Liu, Y. Shi, Source localization in the deep ocean using a convolutional neural network, The Journal of the Acoustical Society of America 147 (4) (2020) EL314–EL319.

[7] H. Niu, Z. Gong, E. Ozanich, P. Gerstoft, H. Wang, Z. Li, Deep-learning source localization using multi-frequency magnitude-only data, The Journal of the Acoustical Society of America 146 (1) (2019) 211–222.

[8] A guide for using the wavelet transform in machine learning (2018). URL `https://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/`

[9] M. Rhif, A. Ben Abbes, B. Martinez, I. R. Farah, An improved trend vegetation analysis for non-stationary ndvi time series based on wavelet transform, Environmental Science and Pollution Research 28 (34) (2021) 46603–46613.

[10] C. Cheng, Principal component analysis (pca) explained visually with zero math (2022).
URL https://towardsdatascience.com/principal-component-analysis-pca-explained-visually-with-zero-math-1cbf392b9e7d

[11] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473 (2014).

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020).

[14] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, C. Feichtenhofer, Multiscale vision transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 6824–6835.

[15] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, C. Feichtenhofer, Mvitv2: Improved multiscale vision transformers for classification and detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 4804–4814.