A differentiable short-time Fourier transform with respect to the window length.

Maxime Leiber
INRIA, DI/ENS, PSL Research University
Safran Tech, Groupe Safran
maxime.leiber@inria.fr

Axel Barrau
Offroad
5 ter rue Parmentier,
94140 Alfortville, France

Yosra Marnissi
Safran Tech, Groupe Safran
Rue des Jeunes Bois-Châteaufort,
78772 Magny-les-Hameaux Cedex, France

Dany Abboud
Safran Tech, Groupe Safran
Rue des Jeunes Bois-Châteaufort,
78772 Magny-les-Hameaux Cedex, France

***Abstract*—In this paper, we revisit the use of spectrograms in neural networks, by making the window length a continuous parameter optimizable by gradient descent instead of an empirically tuned integer-valued hyperparameter. The contribution is mainly theoretical at the moment, but plugging the modified STFT to any existing neural network is straightforward. We first define a differentiable version of the STFT in the case where local bins centers are fixed and independent of the window length parameter. We then discuss the more difficult case where the window length affects the position and number of bins. We illustrate the benefits of this new tool on an estimation and a classification problems, showing it can be of interest not only to neural networks but to any STFT-based signal processing algorithm.**

***Index Terms*—STFT, spectrogram, window length, backpropagation, differentiable optimization, neural networks**

I. Introduction

The short-time Fourier transform (STFT) is an important tool for analyzing non-stationary signals such as transient events like animal sounds [1], discontinuous events like electroencephalography signals [2] as well as smoothly varying multi-harmonic signals typically measured on variable speed mechanical components [3]. Spectrograms built from the outputs of the STFT can be used for simple visualization and understanding of non-stationary signals. In this respect, several researchers have been working to improve the readability of spectrograms using some post-processing techniques such as reassignment [4] and synchro-squeezing [5], [6]. Spectrograms can be also combined with other processing methods to perform more advanced tasks. For instance, [7] combines principal component analysis with vibration signals spectrograms for mechanical fault detection. Spectrograms have also been commonly used to feed transformers, recurrent and convolutional neural networks in various applications, e.g. identification and estimation [8], speech recognition [9], [10], music detection [11], electrocardiogram classification [12], data augmentation [13], source separation [14] etc.

In this work, we are particularly interested in the latter applications i.e those combining neural networks and spectrograms. In this framework, the window length of the spectrogram is usually a fixed parameter, empirically set by trial-and-error or to the default values of commonly used signal processing libraries, without thorough investigation or further justification. But this parameter deserves to be carefully set, as it determines the trade-off between the time and frequency resolution of the spectrogram, in accordance with Heisenberg's uncertainty principle.

Our main contribution in this paper is a new paradigm for window length optimization we believe could become a standard way of tuning the window length of spectrograms used as input to neural networks. It consists in modifying the definition of the STFT operator to make the window length a continuous parameter w.r.t. which spectrogram values can be differentiated. The main idea is to break down the *window size* parameter into two variables: a *numerical window size* integer parameter and a *time resolution* continuous parameter. This distinction has multiple consequences regarding the differentiability of the STFT and, in turn, gives the window length parameter a similar role to neural network weights.In fact, the differentiability proofs and backpropagation formulas provided in this paper allows a joint tuning of the neural network weights and spectrogram resolution (via its window size).

It is worth to note that the problem of searching the best window length is not new and several methods have been proposed in the literature such as [15]–[17]. However, we emphasize that our goal in this paper is not to propose the best way for optimizing the window size parameter, but instead a method for online optimization of the spectrogram resolution in a neural network with gradient algorithms. In that respect, the aforementioned methods are not adapted to our context as they are offline methods. To our knowledge, the only attempt to optimize the window length with a gradient descent is found in the recent work [18] using a Gaussian window and which can be seen as a special case of the theory we propose. However, our paper goes further: the STFT transform is mathematically differentiable and all calculations with propagation and backpropagation formulas are provided.

This paper is organized as follows. In Section II, we start by giving some definitions and notations. In Section III we introduce the modified STFT, being differentiable w.r.t. the window length. In Section IV we extend the theory to the difficult case of STFT with fixed-overlap. In Section V we illustrate the effectiveness of our approach through two applications and we close our discussion in Section VI with few concluding remarks.

II. Definitions and Notations

All over this paper we will refer to the STFT as an operation taking a one-dimensional time series $s[t]$ as input and returning a 2-dimensional table $\mathcal{S}[i, f]$, each column $\mathcal{S}[i, :]$ being the Discrete Fourier Transform (DFT) of a slice of length denoted by $L = 2^n$ of the signal s going from an index b_i to an index b_i+L-1, multiplied by a second sequence h_L called "tapering function". This leads to the following formula we take as a definition of the STFT:

$$\begin{aligned}\mathcal{S}[i, f] &= \mathcal{F}(h_L s[b_i : b_i + L - 1])[f] \\ &= \sum_{k=0}^{L-1} h_L[k] s[b_i + k] e^{-2j\pi k f / L},\end{aligned} \tag{1}$$

where $\mathcal{F}(\cdot)$ denotes the Discrete Fourier Transform (DFT) operator, i is an integer expressing the index of the slice and b_i the associated starting point. Starting indices b_i of time intervals are usually equally spaced, so we only have to set the first index b_0 and spacing Δb between b_i and b_{i+1}. This spacing is usually defined as a percentage of L, through a ratio α called *overlapping*: $\Delta b = \lfloor \alpha L \rfloor$ where $\lfloor \cdot \rfloor$ denotes the integer part. Finally, several choices exist for the tapering function h_L. A common one is the *Hann window* defined as:

$$h_L[k] = \frac{1}{2} - \frac{1}{2}\cos(\frac{2\pi k}{L-1}). \tag{2}$$

Having made notations and terminologies clear, we will move next to the core of this paper: building a STFT whose window length L is a continuous parameter w.r.t. which spectrogram values are differentiable.

III. Differentiable fixed-size STFT

Bulding a differentiable version of STFT means writing a formula similar to Eq. (1) where L becomes a continuous parameter and $\mathcal{S}[i, f]$ is differentiable w.r.t. L. We see that each term of the sum is already differentiable w.r.t. L seen as a continuous parameter. If b_i indices are assumed static the only obstacle is the presence of L as a bound of the sum, which suggests breaking down L into a *numerical window size* N and a *time resolution* θ:

$$\mathcal{S}[i, f] = \sum_{k=0}^{N-1} h_{N,\theta}[k] s[b_i + k] e^{-2j\pi k f / N}. \tag{3}$$

where $h_{N,\theta}$ is a tapering function defined on $[0, N-1]$ but taking non-zero values in the interval $[\frac{N-1-\theta}{2}, \frac{N-1+\theta}{2}]$ where we have:

$$h_{N,\theta}(k) = h_\theta(k + \frac{\theta - N + 1}{2}) \tag{4}$$

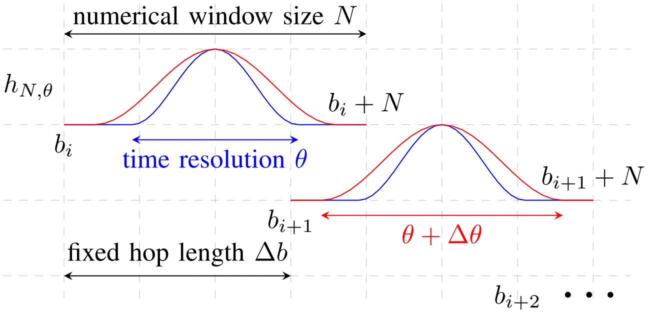

Fig. 1. Fixed-size STFT: on one hand, the size N of the subsignal on which DFT is computed is fixed, on the other hand, the support θ of the tapering function, which actually determines time resolution is allowed to vary. Differentiation will be made w.r.t. this latter parameter.

Let us discuss a little bit the interpretation of N and θ. Parameter θ has a meaning very close to the meaning of L: it sets the time length of the signal slice on which a local spectrum is computed. The difference with classical STFT is that the slice is filled with zeros in order to always be of size N. As a consequence, frequency resolution of the local spectra is not anymore commended by θ but by the second parameter $N > \theta$. This frequency resolution is unrealistic as the signal was padded with zeros: increasing N does not bring more information. Parameter N should only be seen as an upper bound on the time resolution θ making differentiation possible. Now let us compute the differential of our proposed STFT w.r.t. the time resolution θ. We directly obtain:

$$\frac{\partial \mathcal{S}(i, f)}{\partial \theta} = \sum_{k=0}^{N-1} \exp\left(-2j\pi \frac{kf}{N}\right) \frac{\partial h(k)}{\partial \theta} s[b_i + k] \tag{5}$$

where we recognize the STFT of s with tapering function $h' = \frac{\partial h_{N,\theta}(k)}{\partial \theta}$ instead of $h_{N,\theta}$:

$$\frac{\partial \mathcal{S}[s]}{\partial \theta} = \mathcal{S}_{h'}[s] \tag{6}$$

The latter result allows deriving compact gradient backpropagation formulas. We recall gradient backpropagation is the process of computing the derivative of a function $\mathcal{L}$ w.r.t. the input of a function g given its derivative w.r.t. the output of g, i.e., computing $\frac{\mathcal{L}}{I}$ given $\frac{\mathcal{L}}{\mathcal{S}}$ for $\mathcal{S} = g(I)$. In our case we directly obtain:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \theta} &= \sum_{i=1}^{\mathbf{T}} \sum_{f=0}^{N-1} \frac{\partial \mathcal{L}}{\partial \mathcal{S}(i, f)} \frac{\partial \mathcal{S}(i, f)}{\partial \theta} \\ &= \left\langle \frac{\partial \mathcal{L}}{\partial \mathcal{S}}, \mathcal{S}_{h'}[s] \right\rangle,\end{aligned} \tag{7}$$

where $\langle \cdot, \cdot \rangle$ denoted the Froebenius scalar product of two matrices : $\{A, B\} = \sum_{i,j} A_{i,j} B_{i,j}$.

Note we assumed until now that slices positions $[b_i, b_i + k]$ where independent from θ meaning the number of columns of the STFT is constant. This case is useful when the STFT is the input to another algorithm that takes a fixed-size STFT as input, such as neural networks. But this is not representative of all use cases as the standard way to choose the indices b_i is setting an overlapping ratio (usually 50%) then letting indices b_i depend on θ. This more elaborated case is discussed in Section IV below.

IV. Differentiable fixed-overlap STFT

The more usual way of specifying the characteristics of a STFT is defining an overlapping parameter α instead of the bins positions. If window size is a fixed integer L this is straightforward : the gap between starting indices of two slices is simply $\lfloor \alpha L \rfloor$. This is more complicated for differentiable STFT as setting this gap to $\lfloor \alpha\theta \rfloor$ loses differentiability w.r.t. θ while removing the integer part would mean the bound of the interval is not an integer. The solution is setting $b_i = \lfloor \alpha\theta \rfloor$ and compensating the effect of the integer part by the combination of a small shift in the argument of $h_{N,\theta}$ and a factor $e^{2j\pi b_i f/N}$ making the obtained STFT differentiable:

$$\mathcal{S}[i,f] = \sum_{k=0}^{N-1} h_{N,\theta}(k - \mathrm{frac}(i\alpha\theta)) s[b_i + k] e^{\frac{-2j\pi(k+b_i)f}{N}} \quad (8)$$

where $\mathrm{frac} = x - \lfloor x \rfloor$. The remaining of this section will be dedicated to supporting this modified formula, first showing it is continuous w.r.t. θ, then showing it is even differentiable:

Proposition IV.1. *STFT defined by Eq. (8) is everywhere continuous w.r.t. θ.*

Proof. The integer and fractional part functions are continuous everywhere except on integers. So Eq. (8) is continuous everywhere except possibly when $i\alpha\theta$ takes an integer value $p \in \mathbb{N}$. We will prove that it is also continuous then by looking at its left and right limits, i.e. for $\theta = \frac{p}{i\alpha} \pm \epsilon$ with 0 going to zero.

For $\theta = \frac{p}{i\alpha} - \epsilon$, we have $b_i = p-1$ and $\mathrm{frac}(i\alpha\theta)) = 1 - i\alpha\theta$, which implies:

$$\begin{aligned}\mathcal{S}[i,f] &= \sum_{k=0}^{N-1} h_{N,\theta}(k - 1 + i\alpha\theta) s[p-1+k] e^{\frac{-2j\pi(k+p-1)f}{N}} \\ &= \sum_{k=0}^{N-1} h_{N,\theta}(k' + i\alpha\theta) s[p+k'] e^{\frac{-2j\pi(k'+p)f}{N}} \end{aligned} \quad (9)$$

To obtain the second line, we set $k' = k-1$, where k' varies from -1 to $N-2$. But first and last element being zero, the sum is not changed if k' goes from 0 to $N-1$.

For $\theta = \frac{p}{i\alpha} + \epsilon$, we have $b_i = p$ and $\mathrm{frac}(i\alpha\theta)) = i\alpha\theta$, which implies:

$$\mathcal{S}[i,f] = \sum_{k=0}^{N-1} h_{N,\theta}[k - i\alpha\theta] s[p+k] e^{\frac{-2j\pi(k+p)f}{N}} \quad (10)$$

So the limit is the same for $\theta = \frac{p}{i\alpha} \pm \epsilon$ when $\epsilon \to 0$. □

We proved our differentiable fix-overlap STFT is continuous w.r.t. θ. Let us consider its differentiability. The first step is noticing that computations leading to Eq. (7) still hold, h' being replaced by:

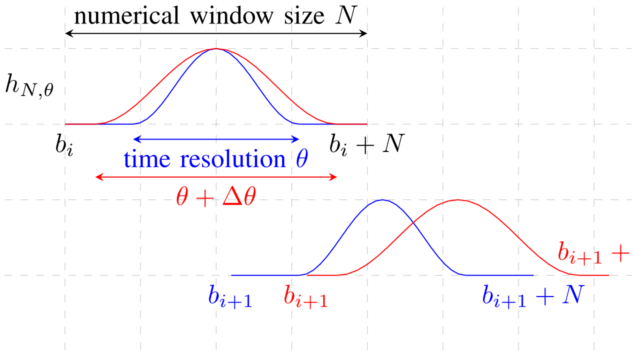

Fig. 2. Differentiable Fixed-overlap STFT: Fig. 2 has similar content to Fig. 1 but illustrates the difference with the fixed-size case: the center of the tapering function vary now with θ and possibly akes a non integer value.

$$h' = \frac{\partial}{\partial\theta}\left[h_{N,\theta}(k - \mathrm{frac}(i\alpha\theta)) e^{2j\pi b_i f/N}\right] \quad (11)$$

where $b_i = \lfloor i\alpha\theta \rfloor$ is a function of θ. Eq. (11) is defined only when $i\alpha\theta$ is not an integer, i.e., when Eq. (8) is obviously differentiable. Extension to any value of θ is the point of Prop. 3.2 below.

Proposition IV.2. *The modified fixed-overlap STFT defined by Eq. (8) is everywhere differentiable w.r.t. θ.*

Proof. If $i\alpha\theta$ is not an integer, we have as in Eq. (5):

$$\frac{\partial\mathcal{S}(i,f)}{\partial\theta} = \sum_{k=0}^{N-1} \tilde{h}(k - \mathrm{frac}(i\alpha\theta)) \partial\theta s[b_i + k] e^{2j\pi(k+p)f/N} \quad (12)$$

with $\tilde{h} = -i\alpha\frac{\partial h_{N,\theta}}{\partial k} + \frac{\partial h_{N,\theta}}{\partial\theta}$ obtain expanding Eq.(11). We recognize the same shape as Eq. (8) meaning a smimilar reasoning will show that the derivative Eq. (12) converges to the same limits when $i\alpha\theta$ goes to an integer by lower or upper values. As a consequence, the proposed fixed-overlap STFT is differentiable everywhere. □

We just proved that we actually obtain the desired differentiable w.r.t. the time resolution parameter θ version of STFT, even in the case of fix-overlap. This allows in particular gradient backpropagation, as Eq. (7) still applies. But fixed-overlap STFT raises a second issue: we assumed the number of columns of the obtained STFT was fixed, which is false as the number of columns increases when the time resolution decreases. The solution is padding the signal with zeros and taking into account all windows covering at least one value of the original signal. Then, the value of each column (replaced with zero when not defined) is a continuous function of θ. This way we obtained a STFT with fixed number of lines but a number of columns which depends on the chosen time resolution, akin to a classical STFT if overlapping ratio is maintained fix while window size changes.

Remark. *In many applications, it is common to consider instead of the complex STFT $\mathcal{S}[i, f]$, the spectrum magnitude $|\mathcal{S}[i, f]|$ also called spectrogram, or any function of $|\mathcal{S}[i, f]|$ such as* $\log|\mathcal{S}[i, f]|$. *As this transformation is not differentiable at zero, a trick is to use a smooth relaxation. A typical example is the smooth hyperbolic function* $\sqrt{|\mathcal{S}[i, f]|^2 + \epsilon}, \epsilon > 0$ *which approaches the absolute value in the limit case* $\epsilon \to 0$.

V. Experiments

In this section, we show how the proposed differentiable STFT can be of an immediate interest for any task involving a spectrogram. The first example illustrates the optimisation of the window length through a simple frequency tracking problem. In the second experiment, we investigate the joint optimisation of the window length with the weights of a simple neural network.

A. Frequency tracking

The objective is to track the instantaneous frequency of a mono-component signal over time. The spectrogram is widely used in the literature to perform such a task. The motivation is that this difficult problem turns to be a peak detection and ridge tracking problem in the spectrogram: the aim is to detect the points of highest energy in the spectrogram plane. The challenge of time-frequency domain estimation is to find the appropriate window length to track fast frequency variations over time while maintaining a good frequency resolution.

In this example, we generate synthetic training data consisting of variable period sine plus white noise. The objective is to find the best window length that achieves the lowest mean square error in this training data

$$\mathcal{L}(\theta) = \frac{1}{J}\sum_{j=1}^{J} \left\|\hat{y}_j^{\theta} - \bar{y}_j\right\|^2 \qquad (13)$$

where $\bar{y}_j$ is the true frequency of the j^{th} signal of the training data and $\hat{y}_j^{\theta}$ is the estimated frequency from the spectrogram of window length θ using some appropriate method. The latter can be a standard signal processing approach or a neural network including or not additional unknown parameters. For simplification reasons, we take in this example, the estimated frequency $\hat{y}[i]$ at each time interval i as the weighted mean of the spectrogram frequencies with weights calculated from the spectrogram amplitude as follows

$$\hat{y}^{\theta}[i] = \frac{\sum_f \mathcal{S}^{\theta}[i, f]f}{\sum_f \mathcal{S}^{\theta}[i, f]} \qquad (14)$$

where $\mathcal{S}^{\theta}$ is the spectrogram of window length θ. The latter is then the only unknown parameter of the loss function.

In our experiments, we consider our fixed size differentiable spectrogram and we run a gradient descend algorithm minimizing (13) allowed by the backpropagation formulas of Sections III and IV. At the convergence of the gradient descent algorithm, the window length has reached the minimum of the loss on Fig. 3. Fig. 4 shows a spectrogram with the estimated window length and Fig. 5 the resulted estimated frequency. We see that the window length looks adapted to the problem although it was automatically tuned through gradient descent. This simple simulation proves the effectiveness of our backpropagation procedure based on a differentiable version of STFT. It also illustrates a general window length tuning methodology applicable to any existing signal processing algorithm involving spectrograms: replace the spectrogram computation step by the computation of our differentiable spectrogram, then optimise the window length by gradient descent based on the backpropagation formulas we have introduced.

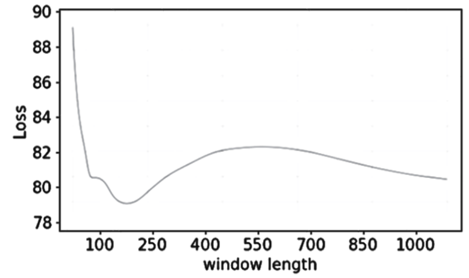

Fig. 3. Loss per window length. At the end of the gradient descent, the time resolution continuous parameter θ reached the minimum value of the above cost function.

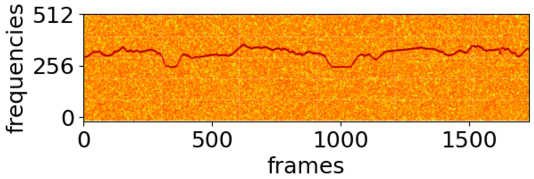

Fig. 4. Spectrogram $\mathcal{S}$ obtained with optimal window length.

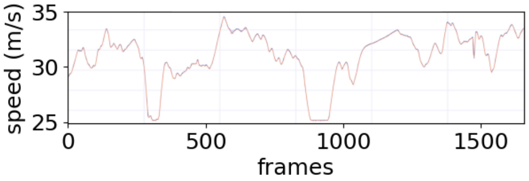

Fig. 5. Frequency estimation $\hat{y}$ based on spectrogram $\mathcal{S}$. The estimation looks close to the true frequency.

B. Joint optimisation with a neural network

This second example illustrates how easy it is to plug our modified STFT into any existing neural network. Indeed, The goal of this experiment is to show that it is possible to jointly optimize the weights of a neural network with the window size of the spectrogram and that the latter will converge towards an optimal window size.

We chose a simple classification task of spoken digit recognition using Free Spoken Digit Dataset (FSDD). The objective is to automatically find the best window size together with the best network weights for the considered dataset that minimizes the cross-entropy with the ground truth. We trained, for several previously fixed window sizes, a 2-layer convolutional neural network in order to compare the accuracy of the network as a function of the window size. We see in Table I, that the window size parameter is very important because the accuracy of the same trained network for different values of the window size varies strongly. Also that, whatever the initial value of the window size is, during a joint optimization with the network weights, the window size parameter converges to an optimal value as displayed on Fig. 6 and 7. Indeed, at the end of the gradient descent, the window length reached the value 34.9 whereas we started the training with a window size of 200.

TABLE I
TRAINING, VALIDATION AND TESTING LOSSES OF CNN PER FIXED WINDOW LENGTH.

window length	train loss	val loss	test loss
10	0.3672	1.1442	1.0391
20	0.0304	0.4443	0.3177
30	0.0089	0.2408	0.0306
40	0.0064	0.2550	0.1249
50	0.0061	0.4859	0.2868

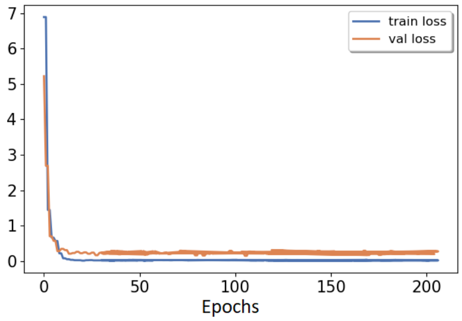

Fig. 6. Training and validation loss per epoch. Losses decrease by jointly optimizing the window length and the weights of the neural network.

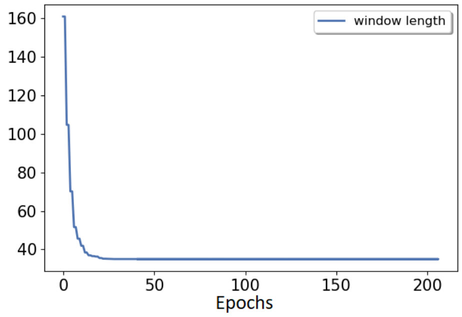

Fig. 7. Window length per epoch. During training, the time resolution continuous parameter θ converged to an optimal value.

VI. CONCLUSION

We presented a modification of STFT making this operation differentiable w.r.t. the window length parameter, and gave the induced backpropagation formulas. As far as we know, this had not been done until now although this algorithm is ubiquitous in signal processing. The main application of this contribution is revisiting the combination of spectrograms and neural networks: instead of directly giving a spectrogram as input to the network, one can now directly give the time signal as input to the network having our differentiable spectrogram as first layer and window length as continuous parameter and let it optimize this window length along with all weights of the network.

REFERENCES

[1] Stafford, K. M., Fox, C. G., and Clark, D. S. (1998). Long-range acoustic detection and localization of blue whale calls in the northeast Pacific Ocean. The Journal of the Acoustical Society of America, 104(6), 3616-3625.

[2] Ramos-Aguilar, R., Olvera-López, J. A., and Olmos-Pineda, I. (2017). Analysis of EEG signal processing techniques based on spectrograms. Research in Computing Science, 145, 151-162.

[3] Leclère, Q., André, H., and Antoni, J. (2016). A multi-order probabilistic approach for Instantaneous Angular Speed tracking debriefing of the CMMNO 14 diagnosis contest. Mechanical Systems and Signal Processing, 81, 375-386.

[4] Flandrin, P., Auger, F., and Chassande-Mottin, E. (2018). Time-frequency reassignment: from principles to algorithms. In Applications in time-frequency signal processing (pp. 179-204). CRC Press.

[5] Thakur, G., Brevdo, E., Fučkar, N. S., and Wu, H. T. (2013). The synchrosqueezing algorithm for time-varying spectral analysis: Robustness properties and new paleoclimate applications. Signal Processing, 93(5), 1079-1094.

[6] Auger, F., Flandrin, P., Lin, Y. T., McLaughlin, S., Meignen, S., Oberlin, T., and Wu, H. T. (2013). Time-frequency reassignment and synchrosqueezing: An overview. IEEE Signal Processing Magazine, 30(6), 32-41.

[7] Yu, H., Guo, Q., Hu, J., and Xu, A. (2006). Rolling bearings fault diagnosis based on adaptive gaussian chirplet spectrogram and independent component analysis. In International Conference on Natural Computation (pp. 321-330).

[8] O'Shea, T. J., Roy, T., and Erpek, T. (2017). Spectral detection and localization of radio events with learned convolutional neural features. In 2017 25th European Signal Processing Conference (EUSIPCO) (pp. 331-335).

[9] Badshah, A. M., Ahmad, J., Rahim, N., and Baik, S. W. (2017). Speech emotion recognition from spectrograms with deep convolutional neural network. In 2017 international conference on platform technology and service (PlatCon) (pp. 1-5).

[10] Gong, Y., Chung, Y. A., and Glass, J. (2021). Ast: Audio spectrogram transformer.

[11] Schlüter, J., and Böck, S. (2014). Improved musical onset detection with convolutional neural networks. In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6979-6983).

[12] Huang, J., Chen, B., Yao, B., and He, W. (2019). ECG arrhythmia classification using STFT-based spectrogram and convolutional neural network. IEEE Access, 7, 92871-92880.

[13] Park, D. S., Chan, W., Zhang, Y., Chiu, C. C., Zoph, B., Cubuk, E. D., and Le, Q. V. (2019). Specaugment: A simple data augmentation method for automatic speech recognition.

[14] Défossez, A. (2021). Hybrid Spectrogram and Waveform Source Separation. arXiv preprint arXiv:2111.03600.

[15] Zhong, J., and Huang, Y. (2010). Time-frequency representation based on an adaptive short-time Fourier transform. IEEE Transactions on Signal Processing, 58(10), 5118-5128.

[16] Kwok, H. K., and Jones, D. L. (2000). Improved instantaneous frequency estimation using an adaptive short-time Fourier transform. IEEE Transactions on Signal Processing, 48(10), 2964-2972.

[17] Czerwinski, R. N., and Jones, D. L. (1997). Adaptive short-time Fourier analysis. IEEE Signal Processing Letters, 4(2), 42-45.

[18] Zhao, A., Subramani, K., and Smaragdis, P. (2021). Optimizing Short-Time Fourier Transform Parameters via Gradient Descent. In 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 736-740).